



(19)中華民國智慧財產局

(12)發明說明書公告本

(11)證書號數：TW I879800 B

(45)公告日：中華民國 114 (2025) 年 04 月 11 日

(21)申請案號：109131301

(22)申請日：中華民國 109 (2020) 年 09 月 11 日

(51)Int. Cl. : H04N19/513 (2014.01)

H04N19/17 (2014.01)

(30)優先權：2019/09/24 歐洲專利局 19306178.5

2019/09/27 歐洲專利局 19290093.4

(71)申請人：法商內數位 C E 專利控股公司(法國) INTERDIGITAL CE PATENT HOLDINGS, SAS
(FR)

法國

(72)發明人：瑞斯 賈甘 RATH, GAGAN (IN)；法蘭賽斯 艾德赫 FRANCOIS, EDOUARD
(FR)；高朋 富蘭克 GALPIN, FRANCK (FR)；博德斯 菲利浦 BORDES,
PHILIPPE (FR)

(74)代理人：陳長文；林嘉興；邵而康

(56)參考文獻：

US 10419754B1

網路文獻 Jianle Chen et al. Algorithm description for Versatile Video Coding and Test Model 6 (VTM 6) Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 15th Meeting: Gothenburg, SE 3-12 July 2019 https://jvet-experts.org/doc_end_user/documents/15_Gothenburg/wg11/JVET-O2002-v2.zip

審查人員：陳哲賢

申請專利範圍項數：28 項 圖式數：22 共 84 頁

(54)名稱

用於視訊編碼或解碼之方法、裝置及器件以及相關的非暫時性電腦可讀媒體及電腦程式產品

(57)摘要

本發明揭示使用判定編碼模式資訊之實施例實施在保持良好效能折衷時使用編碼模式之一組合的靈活性方法。在一個實施例中，作出啟動一第一編碼模式之判定，且在判定是否使用一第二編碼模式時，發生使用該等經判定模式之編碼或解碼。其他實施例實現使用多個參考線、圖框內子區塊分區、矩陣圖框內預測及低頻不可分離變換。

Flexible methods to use a combination of coding modes while keeping good performance tradeoff are implemented with embodiments that determine coding mode information. In one embodiment, determination is made of activation of a first coding mode, and upon determination of whether a second coding mode is used, coding or decoding using the determined modes occurs. Other embodiments enable use of multiple reference lines, intra sub-block partitioning, matrix intra prediction, and low frequency non-separable transforms.

指定代表圖：

符號簡單說明：

300:步驟

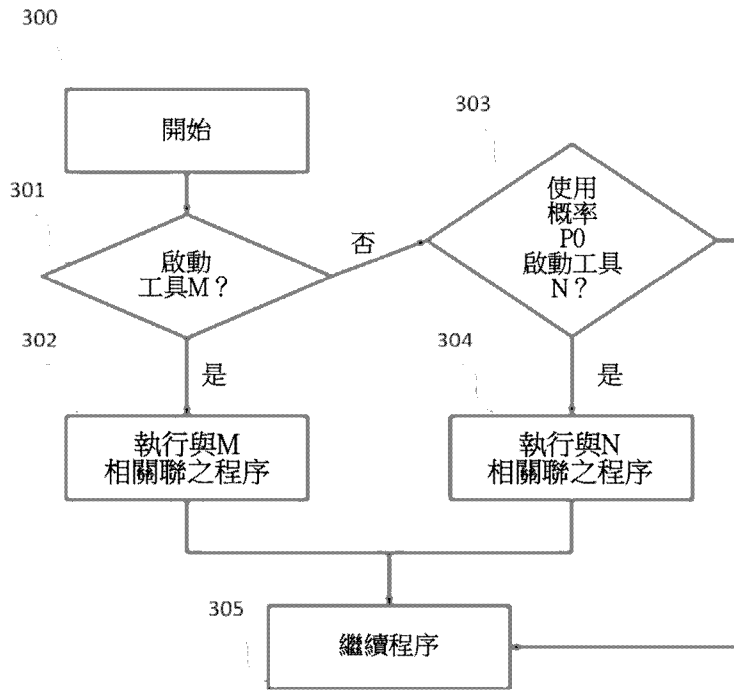
301:步驟

302:步驟

303:步驟

304:步驟

305:步驟



【圖14】



I879800

【發明摘要】

【中文發明名稱】

用於視訊編碼或解碼之方法、裝置及器件以及相關的非暫時性電腦可讀媒體及電腦程式產品

【英文發明名稱】

METHODS, APPARATUS, AND DEVICES FOR VIDEO ENCODING OR DECODING AND RELATED NON-TRANSITORY COMPUTER READABLE MEDIA AND COMPUTER PROGRAM PRODUCTS

【中文】

本發明揭示使用判定編碼模式資訊之實施例實施在保持良好效能折衷時使用編碼模式之一組合的靈活性方法。在一個實施例中，作出啟動一第一編碼模式之判定，且在判定是否使用一第二編碼模式時，發生使用該等經判定模式之編碼或解碼。其他實施例實現使用多個參考線、圖框內子區塊分區、矩陣圖框內預測及低頻不可分離變換。

【英文】

Flexible methods to use a combination of coding modes while keeping good performance tradeoff are implemented with embodiments that determine coding mode information. In one embodiment, determination is made of activation of a first coding mode, and upon determination of whether a second coding mode is used, coding or decoding using the determined modes occurs. Other embodiments enable use of multiple reference lines, intra sub-block partitioning, matrix intra prediction, and low frequency non-separable transforms.

【指定代表圖】

圖14

【代表圖之符號簡單說明】

300:步驟

301:步驟

302:步驟

303:步驟

304:步驟

305:步驟

【發明說明書】

【中文發明名稱】

用於視訊編碼或解碼之方法、裝置及器件以及相關的非暫時性電腦可讀媒體及電腦程式產品

【英文發明名稱】

METHODS, APPARATUS, AND DEVICES FOR VIDEO ENCODING OR DECODING AND RELATED NON-TRANSITORY COMPUTER READABLE MEDIA AND COMPUTER PROGRAM PRODUCTS

【技術領域】

【0001】 本實施例之至少一者大體上係關於一種用於視訊編碼或解碼、壓縮或解壓縮之方法或裝置。

【先前技術】

【0002】 為了達成高壓縮效率，影像及視訊編碼方案通常採用包含運動向量預測之預測，且變換以利用視訊內容中之空間及時間冗餘。一般而言，使用圖框內或圖框間預測來利用圖框內或圖框間相關性，接著變換、量化及熵編碼原始影像與預測影像之間的差異(通常表示為預測誤差或預測殘差)。為了重建視訊，藉由對應於熵編碼、量化、變換及預測之逆程序來解碼壓縮資料。

【發明內容】

【0003】 本實施例之至少一者大體上係關於一種用於視訊編碼或解碼之方法或裝置，且更特定言之，本發明係關於一種用於如在VVC (多功能視訊編碼或H.266)標準中使用特定編碼工具來使用同質編碼語法之方法或裝置。

【0004】 根據一第一態樣，提供一種方法。該方法包括以下步驟：判定是否啟動用於編碼一視訊影像之一部分之一第一編碼工具；判定是否一第二編碼工具經啟動用於編碼該視訊影像之該部分及用於該啟動之條

件；及，有條件地執行該等編碼工具之至少一者。

【0005】 根據一第二態樣，提供一種方法。該方法包含以下步驟：判定是否自一視訊位元串流啟動一第一解碼工具；判定是否一第二解碼工具經啟動及用於該啟動之條件；及，有條件地執行該等解碼工具之至少一者。

【0006】 根據另一態樣，提供一種裝置。該裝置包括一處理器。該處理器可經組態以藉由執行前述方法之任何者來編碼一視訊之一區塊或解碼一位元串流。

【0007】 根據至少一個實施例之另一總體態樣，提供一種器件，其包括：根據解碼實施例之任何者之一裝置；及以下至少一者：(i)一天線，其經組態以接收一信號，該信號包含該視訊區塊；(ii)一頻帶限制器，其經組態以將該所接收之信號限制至包含該視訊區塊之一頻率帶；及(iii)一顯示器，其經組態以顯示代表一視訊區塊之一輸出。

【0008】 根據至少一個實施例之另一總體態樣，提供一種非暫時性電腦可讀媒體，其含有根據所描述之編碼實施例或變體之任何者產生之資料內容。

【0009】 根據至少一個實施例之另一總體態樣，提供一種信號，其包括根據所描述之編碼實施例或變體之任何者產生之視訊資料。

【0010】 根據至少一個實施例之另一總體態樣，一位元串流經格式化以包含根據所描述之編碼實施例或變體之任何者產生之資料內容。

【0011】 根據至少一個實施例之另一總體態樣，提供一種電腦程式產品，其包括指令，當該程式由一電腦執行時，該等指令致使該電腦執行所描述之解碼實施例或變體之任何者。

【0012】 自結合隨附圖式而閱讀之闡釋性實施例之下列實施方式將明白總體態樣之此等及其他態樣、特徵及優點。

【圖式簡單說明】

【0013】 圖1繪示鄰近於一預測區塊之四個參考線之一實例。

【0014】 圖2繪示取決於區塊大小之一子分區。

【0015】 圖3繪示矩陣加權圖框內預測程序。

【0016】 圖4繪示將一CU分為兩個三角預測單元。

【0017】 圖5繪示兩個三角預測單元之間對角線邊緣之加權程序。

【0018】 圖6繪示用於多假設中之4圖框內預測之形狀。

【0019】 圖7繪示低頻不可分離變換(LFNST)程序。

【0020】 圖8繪示SBT之位置、類型及變換類型。

【0021】 圖9繪示所建議之內插濾波器之實例。

【0022】 圖10繪示「IF索引」之一值且「MV」允許導出運動補償濾波器。

【0023】 圖11繪示編碼單元樹語法。

【0024】 圖12繪示一標準、通用、視訊壓縮方案(編碼器)。

【0025】 圖13繪示一標準、通用、視訊解壓縮方案(解碼器)。

【0026】 圖14繪示一模式排除圖。

【0027】 圖15繪示一非排他模式圖。

【0028】 圖16繪示值推斷程序之一概述。

【0029】 圖17繪示一概述傳訊參數。

【0030】 圖18繪示CHP旗標之熵編碼。

【0031】 圖19繪示根據一般描述之態樣之用於編碼/解碼之一基於處

理器之系統。

【0032】 圖20繪示使用所描述之態樣編碼之一方法。

【0033】 圖21繪示使用所描述之態樣解碼之一方法。

【0034】 圖22繪示使用所描述之態樣編碼或解碼之一裝置。

【實施方式】

【0035】 所描述之實施例之上下文在視訊壓縮領域，並解決一編解碼器中編碼模式組合之特定問題。更具體而言，在VVC中，一些模式組合已標準化地自規範移除，因為其等在編碼器上無法提供一良好效能/複雜性折衷。換言之，對一些模式之使用施加取決於上下文不能使用之限制。

【0036】 在一般情況下，所描述之實施例描述一種靈活性方法來使用此等模式之一組合，同時保持良好效能折衷。

【0037】 在下文中，使用涉及本態樣中討論之編碼模式之若干首字母縮寫詞。與各模式相關聯之基礎程序與本說明並非特別相關，而係與啟動彼等模式所應用之限制有關。因此，下文僅提供彼等模式之一簡短說明。可在VVC文文件中找到更多細節。

多參考線(MRL)圖框內預測

【0038】 一區塊之圖框內預測基於區塊之已經重建之鄰近樣本。此等樣本稱為參考樣本。在正常圖框內預測中，僅將編碼區塊之最近行/行用作參考樣本。在多參考線(MRL)圖框內預測中，使用更多圖框內預測之參考線。在圖1中，描繪4個參考線之一實例，其中片段A及F之樣本並非自經重構鄰近樣本獲取，而係分別填充來自片段B及E之最近樣本。

互動MRL-MPM

【0039】 傳訊選定參考線之索引(mrl_idx)並將其用於產生一圖框內預測器。MRL僅適用於MPM(最可能之模式)模式之一者，且不適用於剩餘模式之任何者。使用MRL時，平面及DC模式自圖框內預測模式排除。此係本說明書中解決之模式使用限制之一種情況。

圖框內子分區(ISP)

【0040】 圖框內子分區(ISP)工具將亮度圖框內預測區塊垂直或水平劃分成2個或4個子分區。圖2展示兩種可能性之實例。所有子分區滿足具有至少16個樣本之條件。

【0041】 在ISP中，各子分區之經重構樣本值可用於產生下一個子分區之預測，且各子分區經重複處理。待處理之第一子分區係含有CU之左上角樣本之分區。下一個子分區經向下(水平劃分)或向右(垂直劃分)迭代處理。所有子分區共用相同圖框內模式。

ISP與其他編碼工具之互動。

- 多參考線(MRL)：若一區塊具有除0之外之一MRL索引，則ISP編碼模式將推斷為0且因此ISP模式資訊將不會發送至解碼器。
- MPM用法：在由ISP模式編碼之一區塊中，將MPM旗標推斷為1，且修改MPM清單以排除DC模式並為ISP水平劃分優先設定水平圖框內模式且為垂直劃分優先設定垂直圖框內模式。
- 變換大小限制：所有具有大於16點之一長度之ISP變換都使用DCT-II。
- MTS旗標：若一CU使用ISP編碼模式，則MTS CU旗標將設定為0且不將其發送至解碼器。因此，編碼器將不會對各結果子分區之不同可用變換執行RD測試。ISP模式之變換選擇將代替地為固定的並

根據圖框內模式、處理順序及所利用之區塊大小選擇。因此，無需信令。

矩陣加權圖框內預測(MIP)

【0042】 在矩陣加權圖框內預測(MIP)中，寬度 W 及高度 H 之一矩形區塊之樣本之預測將該區塊之左側之 H 個經重建鄰近邊界樣本之一個行及該區塊上方之 W 個經重建鄰近邊界樣本之一個行作為輸入。預測信號之產生基於以下三個步驟，其等係參考樣本之平均化、平均參考樣本之矩陣向量相乘以及線性內插，如圖3中所展示。平均化對應於參考樣本之一濾波及下採樣。此導致在一1D向量中設定之值。此向量乘以由串流中編碼之一索引選擇之一矩陣，其導致預測區塊之一子採樣版本。執行一最終內插以產生全解析度之預測區塊。

【0043】 對於圖框內模式中之各編碼單元(CU)，發送指示是否將一MIP模式應用於對應預測單元(PU)之一旗標。若應用一MIP模式，則直接編碼MIP模式。

【0044】 所支持之MIP模式之數目取決於區塊大小。

互動MIP-LFNST

【0045】 在大區塊上為MIP啟用LFNST (低頻不可分離變換)。此處，使用對應於平坦模式之LFNST變換。

三角預測模式(TPM)

【0046】 三角預測係一種將預測劃分成兩個三角分區之圖框間預測。如圖4中所展示，一CU在對角線方向或反對角線方向上劃分成兩個三角預測單元(PU)。CU中之各三角預測單元使用自一合併候選者列表導出之其自身運動向量及參考圖框索引進行圖框間預測。

【0047】 將一自適應加權程序應用於兩個三角預測單元之間對角線或反對角線邊緣之預測樣本以導出整個CU之最終預測樣本值，如圖5中所展示。

【0048】 三角預測單元模式僅適用於跳過或合併模式下之CU。當將三角預測單元模式應用於CU時，傳訊將CU劃分成兩個三角預測單元之方向之一索引(此處稱為triangle_idx)加上兩個三角預測單元之運動向量。

互動TMP-SBT

【0049】 三角分區模式不與SBT (子區塊變換)組合使用，即，當經傳訊三角模式等於1時，無需傳訊即將cu_sbt_flag (此處亦稱為「sbt」)推斷為0。

經組合圖框內 - 圖框間預測 (CIIP, Combined Intra-Inter Prediction)

【0050】 經組合圖框內-圖框間預測之一般概念係將在合併模式下執行之一圖框間預測(傳訊合併索引以導出用於運動補償預測之運動資訊)與一圖框內預測模式或另一圖框間模式(例如，單向預測AMVP、跳過或合併)進行組合。最終預測係合併索引預測及藉由圖框內預測模式產生之預測之加權平均值，其中取決於當前樣本與參考樣本之間的圖框內方向及距離應用不同權重。圖6展示用於多假設中之4圖框內預測之形狀。如圖6中所展示，當前區塊經劃分成4個等面積區域。隨著區域遠離圖框內參考樣本，權重逐漸減小。各權重集(表示為 $(w_{intra_i}, w_{inter_i})$)，其中i係自1至4且 $(w_{intra1}, w_{inter1})=(6,2)$ ， $(w_{intra2}, w_{inter2})=(5,3)$ ， $(w_{intra3}, w_{inter3})=(3, 5)$ 且 $(w_{intra4}, w_{inter4})=(2, 6)$ 將應用於一對應區域。當選擇DC或平坦模式，或CU寬度或高度小於4時，將對所有樣本應用相等權重。在多假設CU中進行圖框內預測時，色度分量使用直接模式(相同於

亮度之圖框內方向)。

互動CIIP跳過

【0051】 在跳過模式下不能使用CIIP。

多變換選擇(MTS)

【0052】 多變換選擇(MTS)用於圖框間及圖框內編碼區塊之殘差編碼。除已在HEVC中使用之DCT-II之外，亦使用變換集。新引入之變換矩陣係DST-VII及DCT-VIII。

【0053】 為了控制MTS方案，在SPS級別分別為圖框內及圖框間指定單獨啟用旗標。當在SPS處啟用MTS時，傳訊一CU級別旗標以指示是否應用MTS。此處，MTS僅適用於亮度。

MTS限制

【0054】 當區塊寬度及高度兩者小於或等於32且當CBF旗標(編碼取塊旗標，指示該區塊是否含有殘差資料)等於1時，傳訊MTS CU級別旗標。

【0055】 若MTS CU旗標等於零，則在兩個方向上應用DCT2。然而，若MTS CU旗標等於1，則另外傳訊兩個其他旗標以分別指示水平方向及垂直方向之變換類型。

【0056】 藉由移除圖框內模式及區塊形狀依賴性來使用ISP及隱式MTS之變換選擇。若當前區塊係ISP模式或若當前區塊係圖框內區塊且圖框內及圖框間顯式MTS兩者打開，則僅DST7用於水平及垂直變換核心兩者。

低頻不可分離變換(LFNST)

【0057】 LFNST (低頻不可分離變換)(亦稱為精簡二次變換)應用於

前主變換與量化之間(在編碼器處)以及反量化與逆主變換之間(在解碼器側處)，如圖7中所展示。在LFNST中，根據區塊大小應用4x4不可分離變換或8x8不可分離變換。例如，將4x4 LFNST用於較小區塊(即，最小(寬度,高度)<8)，且將8x8 LFNST用於較大區塊(即，最小(寬度,高度)>4)。

【0058】 下文使用輸入作為一實例描述在LFNST中使用之一不可分離變換之應用。為應用4x4 LFNST，4x4輸入區塊 X

$$X = \begin{bmatrix} X_{00} & X_{01} & X_{02} & X_{03} \\ X_{10} & X_{11} & X_{12} & X_{13} \\ X_{20} & X_{21} & X_{22} & X_{23} \\ X_{30} & X_{31} & X_{32} & X_{33} \end{bmatrix} \quad (3-1)$$

首先表示為一向量 \vec{X} ：

$$[X_{00} \ X_{01} \ X_{02} \ X_{03} \ X_{10} \ X_{11} \ X_{12} \ X_{13} \ X_{20} \ X_{21} \ X_{22} \ X_{23} \ X_{30} \ X_{31} \ X_{32} \ X_{33}]^T \quad (3-2)$$

【0059】 不可分離變換經計算為 $\vec{F} = T \cdot \vec{X}$ ，其中 \vec{F} 指示變換係數向量，且 T 係一16x16變換矩陣。隨後使用該區塊之掃描順序(水平、垂直或對角線)將16x1係數向量 \vec{F} 重新組織為4x4區塊。具較小索引之係數將以較小掃描索引放置於4x4係數區塊中。

【0060】 LFNST中總共有4個變換集且每變換集使用2個不可分離變換矩陣(內核)。自圖框內預測模式至變換集之映射經預定義。若三種交叉分量線性模型模式(CCLM)模式之一者用於當前區塊，則為當前色度區塊選擇變換集0。對於各變換集，由顯式傳訊之LFNST索引進一步指定所選定之不可分離輔助變換候選者。在變換係數之後，每Intra CU在一位元串流中傳訊一次索引。

LNFTS與其他工具之互動

【0061】 LFNST索引經上下文編碼，但不取決於圖框內預測模式，

且僅第一格經上下文編碼。此外，LFNST適用於圖框內及圖框間切片兩者中之圖框內CU，且適用於亮度及色度兩者。

【0062】 當選擇ISP模式時，停用LFNST。當選擇MIP模式時，亦停用LFNST且不傳訊索引。

【0063】 允許LFNST之最大大小限制為64x64。LFNST僅在DCT2中啟用。

子區塊變換(SBT)

【0064】 針對一圖框間預測之CU引入子區塊變換。在此變換模式中，僅殘差區塊之一子部分經編碼用於CU。當cbf之圖框間預測CU等於1時，可傳訊cu_sbt_flag (在所描述之實施例中亦稱為「sbt」)以指示係編碼整個殘差區塊或殘差區塊之一子部分。在前一種情況下，進一步解析圖框間MTS資訊以判定CU之變換類型。在後一種情況下，殘差區塊之一部分通過推斷之自適應變換編碼且殘差區塊之另一部分經清零。

【0065】 當將SBT用於一圖框間編碼之CU時，在位元串流中傳訊SBT類型及SBT位置資訊。如圖8中所指示，存在兩種SBT類型及兩個SBT位置。對於SBT-V (或SBT-H)，TU寬度(或高度)可等於CU寬度(或高度)之一半或CU寬度(或高度)之1/4，導致2:2劃分或1:3/3:1劃分。2:2劃分如同一二進制樹(BT)劃分，而1:3/3:1劃分如同一非對稱二進制樹(ABT)劃分。在ABT劃分中，僅小區域含有非零殘差。若在亮度樣本中一CU之一個維度係8，則不允許沿該維度之1:3/3:1劃分。一CU最多有8種SBT模式。

互動SBT-TMP

【0066】 SBT不適用於採用經組合圖框間-圖框內模式或TPM (三角

分區模式)模式編碼之CU。

可切換濾波器

【0067】 可切換內插濾波器(IF)之原理係藉由選擇用於各區塊預測之IF(**hpellIdx**)來改良運動補償預測。

【0068】 IF之平滑特性通常不同。例如，在多功能視訊編碼建議中，IF索引(**hpellIdx**)按編碼單元(CU)判定且自經編碼「**amvr_precision_idx**」索引導出，該索引指示經編碼運動向量差(MVD)之解析度：若IMV=HALF_PEL，則選擇IF-idx=1，否則選擇IF-idx=0。

【0069】 在合併之情況下，IF索引並非顯式編碼，而係自合併候選者導出。在VVC中，IF索引值可為兩個濾波器(IF-0或IF-1}當中之二者，但IF-1僅可用於HALF_PEL運動向量值。接著若IF索引不等於零且MV水平(或垂直)分量並非HALF_PEL，則使用IF-0 (圖10)。

編碼單元語法樹

【0070】 一編碼單元之語法樹展示於圖11中。樹設計係基於控制編碼模式之不同參數之值。下文給出此等參數之一簡短說明。

【0071】 使用三個主要布林參數：

- 「skip」-布林參數，其指示是否以劃分模式對區塊進行編碼-在跳過模式下，不編碼任何殘差預測信號
- 「merge」-布林參數，其指示是否在合併模式下之圖框間預測中對區塊進行編碼-在合併模式下，運動向量係自候選運動向量列表中導出，且不編碼任何運動向量差
- 「intra」-布林參數，其指示是否以圖框內預測模式對區塊進行編碼

【0072】 取決於此等參數之值，對應於決策樹之四個主要分支之編碼區塊，四個編碼選項係可行的。

選項1-使用跳過模式之圖框間預測

【0073】 當啟用跳過時，將使用跳過模式。跳過模式係一種圖框間預測模式，其中不編碼任何殘差預測信號，且亦無運動向量預測差(其中可添加一校正之mmvd模式中除外)。運動資料(運動向量、參考索引)係自運動向量候選者之清單中預測，運動資料預測變量由一索引識別(此處稱為merge idx)。

【0074】 可使用以下參數。

- 「subblock」-布林參數，其指示是否按子區塊(通常為4x4區塊)導出區塊之運動向量
- 「regular merge」-布林參數，其指示運動向量推導是否將基於合併模式
- 「subblock idx」-對應於子區塊候選索引(仿射候選之ATMVP)之參數
- 「triangle idx」-對應於三角預測模式之類型及候選索引之參數
- 「mmvd」-布林參數，其指示是否選擇具有運動向量差之合併模式(MMVD)
- 「merge idx」-對應於合併模式之運動向量候選列表中之運動向量預測索引之參數
- 「mmvd idx」-對應於MMVD運動向量預測索引之參數

選項2-圖框內預測情況

【0075】 當停用跳過且「圖框內」參數係打開時，將啟動圖框內預

測。當應用圖框內預測時，可使用以下參數：

- 「mip」-布林參數，其指示是否以矩陣加權圖框內預測(MIP)模式對區塊進行編碼
- 「mrl」-布林參數，其指示是否以多參考線(MRL)圖框內預測模式對區塊進行編碼
- 「isp」-布林參數，其指示是否以圖框內子分區(ISP)預測模式對區塊進行編碼
- 「mip_mode」-給予MIP圖框內預測子模式值之參數
- 「mrl_idx」-指示在MRL圖框內預測模式之情況下用於圖框內預測之行之參數
- 「isp_mode」-指示ISP預測之類型之參數
- 「mpm, mpm1 to mpm4」-與最可能模式(MPM)圖框內模式推導相關之旗標
- 「planar」-指示圖框內模式是否為平坦模式之旗標
- 「intradir」-給予圖框內預測方向之參數

選項3-具有合併模式之圖框間預測

【0076】 當停用「skip」及「intra」且「merge」參數係打開時，將啟動具有合併模式之圖框間預測。合併模式係一圖框間預測模式，其中運動向量係自候選運動向量列表導出，且不編碼任何運動向量差（其中可添加一校正之mmvd模式中除外）。一索引經編碼以指示用作預測變量之候選運動向量(此處表示為merge idx)。

【0077】 可使用以下參數：

- 使用在跳過情況中使用之相同參數：「subblock」、「regular

merge」、「subblock idx」、「triangle idx」、「mmvd」、「merge idx」、
「mmvd idx」

- 此外，一個參數限於合併情況：
 - 「ciip」-布林參數，其指示是否選擇經組合圖框間/圖框內預測(CIIP)模式

選項4-使用AMVP模式之圖框間預測

【0078】 當停用「skip」、「intra」及「merge」時，使用具有高級運動向量預測(AMVP)之圖框間預測。AMVP係運動預測之一般概念，包括基於運動向量候選列表執行運動向量之預測。為了識別預測變量，傳訊一索引(此處表示為mvp_idx)。在此情況下，編碼殘差預測資料及運動向量差資訊。

【0079】 當應用圖框內預測時，可使用以下參數。

- 「dir」-指示圖框間預測(單或雙)預測類型之參數
- 「affine」-指示區塊中之運動資料係仿射或平移類型之參數
- 「ref_idx0/1」-指示用於來自參考圖片列表(L0/L1)之時間預測之參考圖片索引之參數
- 「mvd0/1」-與運動向量差相關之參數以添加至運動向量預測
- 「mvp_idx0/1」-指示運動向量候選列表中運動向量預測變量之索引之參數
- 「amvr」係一布林參數，其指示是否將自適應運動向量解析度(AMVR)用於運動補償
- 「gbi」(又名bcw)係一布林參數，其指示是否使用通用雙向預測(GBI)模式

語法

【0080】 下表列出編碼單元及合併語法。

【0081】 可注意，編解碼器中有若干獨占模式，且一些工具組合經停用。

【0082】 可提及以下實例：

-MRL(相關聯之語法元素**intra_luma_ref_idx**)及ISP(相關聯之語法元素**intra_subpartitions_mode_flag**)

-CIIP(相關聯之語法元素**ciip_flag**)及跳過模式(相關聯之語法元素**cu_skip_flag**)

-TPM(沒有相關聯之語法元素，因為其係語法樹之一葉子，若之前解析之所有合併模式為假，則其經啟動)及SBT(相關聯之語法元素**cu_sbt_flag**)

-CIIP及SBT

-LFNST(相關聯之語法元素**lfnst_idx**)及ISP

-當區塊高度或寬度小於16時，LFNST僅用於MIP(相關聯之語法元素**intra_mip_flag**)

-LFNST及MTS(相關聯之語法元素**tu_mts_idx**)

-TPM及AMVP(相關聯之語法元素**general_merge_flag**為假)

-CIIP及AMVP

編碼單元語法

【0083】 自VVC：

coding_unit(x0, y0, cbWidth, cbHeight, cqtDepth, treeType, modeType) {	描述符
chType = treeType == DUAL_TREE_CHROMA ? 1 : 0	
if(slice_type != I sps_ibc_enabled_flag sps_palette_enabled_flag) {	
if(treeType != DUAL_TREE_CHROMA && !((cbWidth == 4 && cbHeight == 4) modeType == MODE_TYPE_INTRA) && !sps_ibc_enabled_flag)	
cu_skip_flag[x0][y0]	ae(v)
if(cu_skip_flag[x0][y0] == 0 && slice_type != I && !(cbWidth == 4 && cbHeight == 4) && modeType == MODE_TYPE_ALL)	
pred_mode_flag	ae(v)
if(((slice_type == I && cu_skip_flag[x0][y0] == 0) (slice_type != I && (CuPredMode[chType][x0][y0] != MODE_INTRA (cbWidth == 4 && cbHeight == 4 && cu_skip_flag[x0][y0] == 0)))) && cbWidth <= 64 && cbHeight <= 64 && modeType != MODE_TYPE_INTER && sps_ibc_enabled_flag && treeType != DUAL_TREE_CHROMA)	
pred_mode_ibc_flag	ae(v)
if((((slice_type == I (cbWidth == 4 && cbHeight == 4) sps_ibc_enabled_flag) && CuPredMode[x0][y0] == MODE_INTRA) (slice_type != I && !(cbWidth == 4 && cbHeight == 4) && !sps_ibc_enabled_flag && CuPredMode[x0][y0] != MODE_INTRA)) && sps_palette_enabled_flag && cbWidth <= 64 && cbHeight <= 64 && && cu_skip_flag[x0][y0] == 0 && modeType != MODE_INTER)	
pred_mode_plt_flag	ae(v)
}	
if(CuPredMode[chType][x0][y0] == MODE_INTRA CuPredMode[chType][x0][y0] == MODE_PLT) {	
if(treeType == SINGLE_TREE treeType == DUAL_TREE_LUMA) {	
if(pred_mode_plt_flag) {	
if(treeType == DUAL_TREE_LUMA)	
palette_coding(x0, y0, cbWidth, cbHeight, 0, 1)	
else /* SINGLE_TREE */	
palette_coding(x0, y0, cbWidth, cbHeight, 0, 3)	
} else {	
if(sps_bdpcm_enabled_flag && cbWidth <= MaxTsSize && cbHeight <= MaxTsSize)	
intra_bdpcm_flag	ae(v)
if(intra_bdpcm_flag)	
intra_bdpcm_dir_flag	ae(v)
else {	
if(sps_mip_enabled_flag && (Abs(Log2(cbWidth) - Log2(cbHeight)) <= 2) && cbWidth <= MaxTbSizeY && cbHeight <= MaxTbSizeY)	
intra_mip_flag[x0][y0]	ae(v)
if(intra_mip_flag[x0][y0])	
intra_mip_mode[x0][y0]	ae(v)
else {	
if(sps_mrl_enabled_flag && ((y0 % CtbSizeY) > 0))	
intra_luma_ref_idx[x0][y0]	ae(v)

if(sps_isp_enabled_flag && intra_luma_ref_idx[x0][y0] == 0 && (cbWidth <= MaxTbSizeY && cbHeight <= MaxTbSizeY) && (cbWidth * cbHeight > MinTbSizeY * MinTbSizeY))	
intra_subpartitions_mode_flag [x0][y0]	ae(v)
if(intra_subpartitions_mode_flag[x0][y0] == 1)	
intra_subpartitions_split_flag [x0][y0]	ae(v)
if(intra_luma_ref_idx[x0][y0] == 0)	
intra_luma_mpm_flag [x0][y0]	ae(v)
if(intra_luma_mpm_flag[x0][y0]) {	
if(intra_luma_ref_idx[x0][y0] == 0)	
intra_luma_not_planar_flag [x0][y0]	ae(v)
if(intra_luma_not_planar_flag[x0][y0])	
intra_luma_mpm_idx [x0][y0]	ae(v)
} else	
intra_luma_mpm_remainder [x0][y0]	ae(v)
}	
}	
}	
if((treeType == SINGLE_TREE treeType == DUAL_TREE_CHROMA) && ChromaArrayType != 0) {	
if(pred_mode_plt_flag && treeType == DUAL_TREE_CHROMA)	
palette_coding(x0, y0, cbWidth / SubWidthC, cbHeight / SubHeightC, 1, 2)	
else {	
if(CclmEnabled)	
cclm_mode_flag	ae(v)
if(cclm_mode_flag)	
cclm_mode_idx	ae(v)
Else	
intra_chroma_pred_mode	ae(v)
}	
} else if(treeType != DUAL_TREE_CHROMA) { /* MODE_INTER or MODE_IBC */	
if(cu_skip_flag[x0][y0] == 0)	
general_merge_flag [x0][y0]	ae(v)
if(general_merge_flag[x0][y0]) {	
merge_data(x0, y0, cbWidth, cbHeight, chType)	
} else if(CuPredMode[chType][x0][y0] == MODE_IBC) {	
mvd_coding(x0, y0, 0, 0)	
if(MaxNumIbcMergeCand > 1)	
mvp_l0_flag [x0][y0]	ae(v)
if(sps_amvr_enabled_flag && (MvdL0[x0][y0][0] != 0 MvdL0[x0][y0][1] != 0)) {	
amvr_precision_idx [x0][y0]	ae(v)
}	
} else {	
if(slice_type == B)	
inter_pred_idc [x0][y0]	ae(v)

if(sps_affine_enabled_flag && cbWidth >= 16 && cbHeight >= 16) {	
inter_affine_flag [x0][y0]	ae(v)
if(sps_affine_type_flag && inter_affine_flag[x0][y0])	
cu_affine_type_flag [x0][y0]	ae(v)
}	
if(sps_smvd_enabled_flag && !mvd_l1_zero_flag && inter_pred_idc[x0][y0] == PRED_BI && !inter_affine_flag[x0][y0] && RefIdxSymL0 > -1 && RefIdxSymL1 > -1)	
sym_mvd_flag [x0][y0]	ae(v)
if(inter_pred_idc[x0][y0] != PRED_L1) {	
if(NumRefIdxActive[0] > 1 && !sym_mvd_flag[x0][y0])	
ref_idx_l0 [x0][y0]	ae(v)
mvd_coding(x0, y0, 0, 0)	
if(MotionModelIdc[x0][y0] > 0)	
mvd_coding(x0, y0, 0, 1)	
if(MotionModelIdc[x0][y0] > 1)	
mvd_coding(x0, y0, 0, 2)	
mvp_l0_flag [x0][y0]	ae(v)
} else {	
MvdL0[x0][y0][0] = 0	
MvdL0[x0][y0][1] = 0	
}	
if(inter_pred_idc[x0][y0] != PRED_L0) {	
if(NumRefIdxActive[1] > 1 && !sym_mvd_flag[x0][y0])	
ref_idx_l1 [x0][y0]	ae(v)
if(mvd_l1_zero_flag && inter_pred_idc[x0][y0] == PRED_BI) {	
MvdL1[x0][y0][0] = 0	
MvdL1[x0][y0][1] = 0	
MvdCpL1[x0][y0][0][0] = 0	
MvdCpL1[x0][y0][0][1] = 0	
MvdCpL1[x0][y0][1][0] = 0	
MvdCpL1[x0][y0][1][1] = 0	
MvdCpL1[x0][y0][2][0] = 0	
MvdCpL1[x0][y0][2][1] = 0	
} else {	
if(sym_mvd_flag[x0][y0]) {	
MvdL1[x0][y0][0] = -MvdL0[x0][y0][0]	
MvdL1[x0][y0][1] = -MvdL0[x0][y0][1]	
} else	
mvd_coding(x0, y0, 1, 0)	
if(MotionModelIdc[x0][y0] > 0)	
mvd_coding(x0, y0, 1, 1)	
if(MotionModelIdc[x0][y0] > 1)	
mvd_coding(x0, y0, 1, 2)	
mvp_l1_flag [x0][y0]	ae(v)
}	
} else {	
MvdL1[x0][y0][0] = 0	

MvdL1[x0][y0][1] = 0	
}	
if((sps_amvr_enabled_flag && inter_affine_flag[x0][y0] == 0 && (MvdL0[x0][y0][0] != 0 MvdL0[x0][y0][1] != 0 MvdL1[x0][y0][0] != 0 MvdL1[x0][y0][1] != 0)) (sps_affine_amvr_enabled_flag && inter_affine_flag[x0][y0] == 1 && (MvdCpL0[x0][y0][0][0] != 0 MvdCpL0[x0][y0][0][1] != 0 MvdCpL1[x0][y0][0][0] != 0 MvdCpL1[x0][y0][0][1] != 0 MvdCpL0[x0][y0][1][0] != 0 MvdCpL0[x0][y0][1][1] != 0 MvdCpL1[x0][y0][1][0] != 0 MvdCpL1[x0][y0][1][1] != 0 MvdCpL0[x0][y0][2][0] != 0 MvdCpL0[x0][y0][2][1] != 0 MvdCpL1[x0][y0][2][0] != 0 MvdCpL1[x0][y0][2][1] != 0)) {	
amvr_flag[x0][y0]	ae(v)
if(amvr_flag[x0][y0])	
amvr_precision_idx[x0][y0]	ae(v)
}	
if(sps_bcw_enabled_flag && inter_pred_idc[x0][y0] == PRED_BI && luma_weight_l0_flag[ref_idx_l0 [x0][y0]] == 0 && luma_weight_l1_flag[ref_idx_l1 [x0][y0]] == 0 && chroma_weight_l0_flag[ref_idx_l0 [x0][y0]] == 0 && chroma_weight_l1_flag[ref_idx_l1 [x0][y0]] == 0 && cbWidth * cbHeight >= 256)	
bcw_idx[x0][y0]	ae(v)
}	
}	
if(CuPredMode[chType][x0][y0] != MODE_INTRA && !pred_mode_plt_flag && general_merge_flag[x0][y0] == 0)	
cu_cbf	ae(v)
if(cu_cbf) {	
if(CuPredMode[chType][x0][y0] == MODE_INTER && sps_sbt_enabled_flag && !ciip_flag[x0][y0] && !MergeTriangleFlag[x0][y0]) {	
if(cbWidth <= MaxSbtSize && cbHeight <= MaxSbtSize) {	
allowSbtVerH = cbWidth >= 8	
allowSbtVerQ = cbWidth >= 16	
allowSbtHorH = cbHeight >= 8	
allowSbtHorQ = cbHeight >= 16	
if(allowSbtVerH allowSbtHorH allowSbtVerQ allowSbtHorQ)	
cu_sbt_flag	ae(v)
}	
if(cu_sbt_flag) {	
if((allowSbtVerH allowSbtHorH) && (allowSbtVerQ allowSbtHorQ))	
cu_sbt_quad_flag	ae(v)
if((cu_sbt_quad_flag && allowSbtVerQ && allowSbtHorQ) (!cu_sbt_quad_flag && allowSbtVerH && allowSbtHorH))	
cu_sbt_horizontal_flag	ae(v)
cu_sbt_pos_flag	ae(v)
}	
}	
LfstDcOnly = 1	
LfstZeroOutSigCoeffFlag = 1	
transform_tree(x0, y0, cbWidth, cbHeight, treeType)	

lfnstWidth = (treeType == DUAL_TREE_CHROMA) ? cbWidth / SubWidthC : cbWidth	
lfnstHeight = (treeType == DUAL_TREE_CHROMA) ? cbHeight / SubHeightC : cbHeight	
if(Min(lfnstWidth, lfnstHeight) >= 4 && sps_lfnst_enabled_flag == 1 && CuPredMode[chType][x0][y0] == MODE_INTRA && IntraSubPartitionsSplitType == ISP_NO_SPLIT && (!intra_mip_flag[x0][y0] Min(lfnstWidth, lfnstHeight) >= 16) && tu_mts_idx[x0][y0] == 0 && Max(cbWidth, cbHeight) <= MaxTbSizeY) {	
if(LfnstDcOnly == 0 && LfnstZeroOutSigCoeffFlag == 1)	
lfnst_idx[x0][y0]	ae(v)
}	
}	

合併資料語法

【0084】 自VVC章節7.3.8.7

	描述符
merge_data(x0, y0, cbWidth, cbHeight, chType) {	
if (CuPredMode[chType][x0][y0] == MODE_IBC) {	
if(MaxNumIbcMergeCand > 1)	
merge_idx[x0][y0]	ae(v)
} else {	
if(MaxNumSubblockMergeCand > 0 && cbWidth >= 8 && cbHeight >= 8)	
merge_subblock_flag[x0][y0]	ae(v)
if(merge_subblock_flag[x0][y0] == 1) {	
if(MaxNumSubblockMergeCand > 1)	
merge_subblock_idx[x0][y0]	ae(v)
} else {	
if((cbWidth * cbHeight) >= 64 && ((sps_ciip_enabled_flag && cu_skip_flag[x0][y0] == 0 && cbWidth < 128 && cbHeight < 128) (sps_triangle_enabled_flag && MaxNumTriangleMergeCand > 1 && slice_type == B)))	
regular_merge_flag[x0][y0]	ae(v)
if(regular_merge_flag[x0][y0] == 1) {	
if(sps_mmvd_enabled_flag)	
mmvd_merge_flag[x0][y0]	ae(v)
if(mmvd_merge_flag[x0][y0] == 1) {	
if(MaxNumMergeCand > 1)	
mmvd_cand_flag[x0][y0]	ae(v)
mmvd_distance_idx[x0][y0]	ae(v)
mmvd_direction_idx[x0][y0]	ae(v)
} else {	
if(MaxNumMergeCand > 1)	
merge_idx[x0][y0]	ae(v)
}	
} else {	
if(sps_ciip_enabled_flag && sps_triangle_enabled_flag && MaxNumTriangleMergeCand > 1 && slice_type == B && cu_skip_flag[x0][y0] == 0 && (cbWidth * cbHeight) >= 64 && cbWidth < 128 && cbHeight < 128) {	

ciip_flag [x0][y0]	ae(v)
if(ciip_flag[x0][y0] && MaxNumMergeCand > 1)	
merge_idx [x0][y0]	ae(v)
if(!ciip_flag[x0][y0] && MaxNumTriangleMergeCand > 1) {	
merge_triangle_split_dir [x0][y0]	ae(v)
merge_triangle_idx0 [x0][y0]	ae(v)
merge_triangle_idx1 [x0][y0]	ae(v)
}	
}	
}	
}	
}	

推斷模式

【0085】 除此等模式排除之外，在編解碼器中，自動推斷一些模式或值。例如：

-在IBC模式之情況下，將半採樣內插濾波器索引**hpellIdx**設定為等於0。在其他情況下，自AMVR推導**hpellIdx**(自適應運動向量解析度，**amvr_resolution_idx**)。若**amvr_resolution_idx**等於Half-Pixel-resolution，則將**hpellIdx**設定成等於1，否則將其設定為0。若最終(經重建)運動向量係一半像素位置且**hpellIdx**等於1，則內插濾波器IF-1用於運動補償，否則使用內插濾波器(IF-0，預設設定)。

$$hpellIdx = (amvr_resolution_idx == HalfPelResolution) ? 1 : 0$$

-MRL及非MPM(最可能模式)模式：若MRL索引不為0，則推導該模式在MPM清單中(不能設定任意圖框內模式)。

-CIIP圖框內模式：圖框內部分始終使用平坦模式(不能設定任意圖框內模式)。

-當使用ISP或SBT時MTS(多變換集)減少

-在MMVD(合併運動向量差)之情況下，合併索引：僅可設定索引0或1，其對應於第一或第二合併候選者。不能設定其他合併候選者。

使用CABAC之熵編碼

【0086】 在HEVC及VVC中，一基於上下文之二進制算術編碼器(CABAC)可用於熵編碼。存在另一編碼模式，稱為「旁路」編碼，此處不予討論。用於編碼一位元之旁路模式包括簡單地編碼位元串流中之位元值。

【0087】 為編碼/解碼一語法元素，CABAC如下工作：

- 語法元素首先經二進制化為一連串格；
- 接著對於各格
 - 選擇一上下文CTX；其與一上下文模型相關聯，該模型儲存「1」或「0」之格之上下文概率；此上下文概率對應於格知道上下文值CTX取得值b之經估計條件概率 $P(b/CTX)$ ；
 - 選擇基於上下文資訊，諸如當前待編碼之符號(語法元素)附近之已編碼符號(語法元素)、CU/PU/TU大小或形狀、切片類型。
 - 根據上下文概率 $P(b/CTX)$ 使用一算術編碼器編碼/解碼格；
 - 一旦編碼，上下文概率 $P0=P(0/CTX)$ 或 $P1=P(1/CTX)$ 經更新。

【0088】 對於算術編碼器，概率 $P0/P1$ 之值(及初始值)對所產生之位元數目具有一重要影響。例如，若以低值之一概率 $P1$ 編碼格1，則與概率 $P1$ 之高值相比，其將需要更多位元。

【0089】 在上文所描述之情況下，儘管因為其減少模式之間的組合而降低編碼器之複雜性(當兩種模式僅可排他使用時，編碼器僅需單獨測試各模式之使用；不做經組合之兩種模式之測試)，但可藉由移除某些靈活性而影響編解碼器之原始效能。

【0090】 所描述之實施例建議使用語法改變、特設熵上下文、熵初始概率參數及編碼選擇之一組合將上文硬約束變換為較軟約束。

【0091】 如上文所述，通常自編碼選擇移除不良複雜性/增益權衡組合或模式選擇。

【0092】 所描述之態樣提供一種藉由在使用一低複雜性編碼器時保持相同效能(即，複雜性/增益折衷係相同的)或藉由使用自適應語法、特設上下文、CABAC參數及編碼方法使此等模式用於更高複雜性編碼器來增加編解碼器靈活性之方法。在此情況下，編碼器更複雜，但期望增益更高。例如，其對於非即時編碼器(諸如在視訊點播應用中)有用，其中視訊串流經編碼一次，而無即時傳遞之約束。

【0093】 吾人建議使用以下之一組合將說明書中之上文硬約束變換為較軟約束：

- 語法改變(例如，取決於另一參數移除或改變語法元素)
- 特設熵上下文，
- 熵初始概率參數
- 編碼選擇。

模式排除

【0094】 在圖14及圖15中，吾人展示使用兩個編碼工具M及N自硬排除變為軟排除之程序中之整個變化。即使使用一相關聯概率模型P1啟動編碼工具M，編碼工具N仍可用(關於細節，參見下文)。

【0095】 圖14繪示「模式排除」情況。其對應於先前技術，其中兩個模式M及N係互斥的。步驟300對應於整個程序之開始。在步驟301中，檢查工具M之啟動(例如，藉由測試一旗標)。

【0096】 若工具M經啟動，則在步驟302中執行工具M。接著，程序進行至繼續整個程序之步驟305。

【0097】 若未啟動工具M，則檢查工具N之啟動(步驟303)。界定啟動工具N之一概率P0。若工具N經啟動，則在步驟304中執行工具N。接著，程序進行至繼續整個程序之步驟305。若未啟動工具N，則程序進行至繼續整個程序之步驟305。

【0098】 在此情況下，僅在未啟動工具M時才可啟動工具N。模式M及N之使用係排他的。

【0099】 圖15繪示「非排他模式」情況。其對應於所描述之實施例之一個態樣，其中模式M及N可經非排他性地使用。步驟400對應於整個程序之開始。在步驟401中，檢查工具M之啟動。

【0100】 若工具M經啟動，則在步驟402中執行工具M。接著在步驟403中檢查工具N之啟動。界定啟動工具N之概率P1。若工具N經啟動，則在步驟404中執行工具N。接著，程序進行至整個程序之步驟406。若未啟動工具N，則程序進行至繼續整個程序之步驟406。

【0101】 若未啟動工具M，則檢查工具N之啟動(步驟405)。界定啟動工具N之概率P0。若工具N經啟動，則在步驟403中執行工具N。接著，程序進行至繼續整個程序之步驟406。若未啟動工具N，則程序進行至繼續整個程序之步驟406。

【0102】 在此情況下，在啟動工具M或未啟動工具M之兩種情況下，可啟動工具N。模式M及N之使用並非排他的。

模式推斷

【0103】 在圖16及圖17中，吾人展示具有相關聯概率之一推斷值至

一傳訊值之整體變換(關於細節，參見下文)。此處考慮兩個工具M及N，且工具N由一參數控制。

【0104】 圖16繪示「值推斷」情況。其對應於其中推斷工具N之參數(未編碼)之先前技術。步驟500對應於整個程序之開始。在步驟501中，檢查工具M之啟動。

【0105】 若工具M經啟動，則在步驟502中檢查工具N之啟動條件。若工具N之啟動條件為真，則在步驟503中以一參數值v1執行工具N。接著，程序進行至繼續整個程序之步驟505。若工具N之啟動條件為假，則在步驟504中以一參數值v0執行工具N。接著，程序進行至繼續整個程序之步驟505。圖中未展示工具M程序在工具N之前或之後執行。

【0106】 在此情況下，總是推斷與工具N相關聯之參數。

【0107】 圖17繪示「傳訊參數」情況。其對應於其中工具N之參數經明確地傳訊之所描述之實施例之一個態樣。步驟600對應於整個程序之開始。在步驟601中，檢查工具M之啟動。

【0108】 若工具M經啟動，則在步驟602中檢查工具N之啟動條件。若工具N之啟動條件為真，則檢查工具N之參數值(且可能自串流解碼)。若參數為v0，則在步驟605中以參數值v0啟動工具N。接著，程序進行至繼續整個程序之步驟606。否則，在步驟604中以參數值v1啟動工具N。接著，程序進行至繼續整個程序之步驟606。

【0109】 若未啟動工具M，則在步驟605中以參數值v0啟動工具N。接著，程序進行至繼續整個程序之步驟606。圖中未展示工具M程序在工具N之前或之後執行。

【0110】 在此情況下，不推斷與工具N相關聯之參數，而係自應用

於工具N之特定條件對其進行解碼或推導。

語法改變

【0111】 第一改變係語法上之一改變，其覆蓋模式排除及/或參數推斷態樣。

【0112】 關於模式排除態樣，移除模式排除。模式N之啟動不再依賴於模式M之啟動。工具N之啟動條件經變換為一特設新CABAC上下文。

【0113】 關於推斷值態樣，引入一新語法元素以顯式編碼值。新語法元素使用一或多個新特設CABAC上下文來替換參數推斷。

【0114】 在一變體中，模式排除態樣(移除模式排除)及/或推斷值態樣(用於顯式編碼該值之新語法元素)在一編碼參數之一條件下(較佳在高級語法中，例如，在切片頭、圖片參數集(PPS)、序列參數集(SPS)、自適應參數集(APS)或解碼參數集中編碼)進行控制/啟動。

CABAC上下文改變

模式排除

【0115】 由於使用一熵編碼器(在此情況下為CABAC)來編碼工具N之使用，因此在啟動工具M時使用一不同格。需要一新CABAC上下文來區分其中模式M是否經啟動之情況。為與模式N相關聯之語法元素之CABAC上下文導出添加模式M啟動之一條件。第一格之編碼(在未啟動模式M時)保持不變且使用相同概率模型P0 (初始值相同)。

值推斷

【0116】 由於使用一熵編碼器(在此情況下為CABAC)來編碼與工具N相關聯之值V0或V1。與值V之編碼相關聯之概率模型P1 (等於V0或V1)

設定為諸如其具有V1 (先前推斷之值)之一高概率。

CABAC初始值自適應

模式排除

【0117】 現描述如何設定新概率模型P1。為不影響選擇在使用工具M時始終排除工具N之一編碼器之編碼效能，初始化此格之概率係低的。

【0118】 實際上，若很少使用一模式，但在CABAC模型中使用之相關聯概率係高的，則將使用很多位元來傳訊該模式未使用。另一方面，若概率低，則與不使用此模式之一編解碼器相比，不使用該模式將對速率沒有影響/影響很小。

【0119】 在下文中，吾人解釋CABAC初始化程序改變之基本原理。

【0120】 在VVC中，取決於QP (量化參數)之一線性模型給出CABAC模型之初始概率值，該線性模型由兩個參數a (乘積因數或斜率)及b (偏移)控制。斜率a及偏移b兩者在說明書中針對對應於一語法元素之一給定格進行硬編碼。隨著CABAC更新各讀取格之概率，亦使用2個「視窗大小」來更新符號概率。取決於視窗大小，概率更新將為快或慢。

【0121】 例如，CIIP旗標參數如下：a=0.024且b=0.29 (此處，為清楚起見，吾人將係數表示為浮點值)。接著取決於QP給出CABAC之初始概率使用：

$$\text{proba}(\text{QP})=(0.024/2)*(\text{QP}-16)+0.29$$

實際上，對於34之一QP之概率大致為0.5。

【0122】 為了不懲罰在啟動模式M時不會測試模式N之一低複雜性編碼器，吾人建議使用a及b較小之一低概率模型初始化CABAC概率。因

此，指示不使用模式N之格之編碼(例如格0)將需要很少位元，因為具有一格1之概率低，且相反，具有一格0之概率高。例如： $a=0$ 且 $b=0.008$ 。在當前VTM-6.0中，使用當前斜率及偏移編碼，其對應於CABAC之一初始化參數32。可使用保持初始概率低之其他值。例如，保持QP之初始概率函數： $a=0.0117$ $b=0.008$ 。在當前VTM-6.0中，使用當前斜率及偏移編碼，其對應於CABAC之一初始化參數40。

【0123】 藉由使用此等初始概率模型，一低複雜性編碼器(如VVC中所描述之編碼器)在啟動N時不執行模式N不會造成效能損失(效能保持大致相同)。但，若需要，一更複雜編碼器可利用新可用模式。實際上，此一編碼器可測試更多模式組合，其在大多數情況下就壓縮增益而言係有益的。

【0124】 在一變體中，與新語法元素相關聯之視窗大小經縮短以便更快地適應概率模型(關於細節，參見下文)。

值推斷

【0125】 概率初始化原理與上文相同：將一較低概率設定為先前未推斷之值。

變體

【0126】 若新引入之模式或非推斷值之邏輯經反轉，則概率初始值將設定為接近1之一值(而非接近0)。

編碼方法

【0127】 為了提高高複雜性編碼器之效能，需要將使用新引入模式組合/值之概率校正為其正確預期值。然而，由於概率模型已針對編碼器/解碼器固定(參見上文參數a、b及視窗大小)，因此選擇新模式之初始成本

對於選擇新模式而言可能太高。在評估啟動模式N之成本時，由於其概率具有一低值，因此其成本較高，因此對模式N之懲罰很大且因此選擇機會很少。在此一情況下，由於很少選擇模式N，因此使用其之概率始終很小，且幾乎沒有使用其之機會。

【0128】 為了在不改變初始值之情況下改變與模式相關聯之概率，一可行編碼方法包括：在前幾個區塊啟動模式M時，使用模式N計算模式N之一偏置RD (速率失真)成本：

$$C=D+\lambda(R-\alpha)$$

其中D係失真，R係比率且 λ 係經典RD成本估計中之比率失真參數。 α 係在啟動模式M時計算為偏好模式N之一小數位元成本(正)。例如，可精確地計算因數 α ，諸如 $(R-\alpha)$ 等於1位元，其對應於當前編碼格之一概率0.5。因此，即使模式N之概率很小，其亦不會受到懲罰，且與不使用該模式時相比，只要編碼器提供一更佳失真值，即可由編碼器選擇。在處理連續區塊時，概率模型逐漸收斂至使用模式N之實際概率。

【0129】 在處理足夠數目個區塊之後，即，當概率模型收斂至對真實模式使用概率之一良好估計時，為了在RD成本公式中使用實際比率成本R，將因數 α 降低至0。可藉由評估概率值P0或P1之變化來評估概率模型之穩定性，同時編碼格。若變化低於一給定臨限值(例如0.01)，則該模型視作穩定的且可開始減小 α 。替代地，一旦處理一給定數目個區塊(例如20)，即可決定 α 之減小。

【0130】 區塊自適應之數目取決於編碼器/解碼器中固定之視窗大小。視窗大小越短，自適應越快。

硬模式至軟模式改變之實例

跳過模式下之CIIP模式

【0131】 此處，模式M為跳過(即無殘差之合併)且模式N係CIIP模式。第一改變涉及編碼單元資料中之語法：啟用CIIP之`cu_skip_flag`不再依賴於`cu_skip_flag` (其表示跳過模式啟動)。此外，為了避免在未啟動之CIIP及三角之情況下進行無用讀取，亦調整讀取條件或不讀取常規合併模式旗標之條件。

<pre>if((cbWidth * cbHeight) >= 64 && (sps_ciip_enabled_flag && cu_skip_flag[x0][y0] == 0 && cbWidth < 128 && cbHeight < 128) (sps_triangle_enabled_flag && MaxNumTriangleMergeCand > 1 && slice_type == B))</pre>	
<code>regular_merge_flag[x0][y0]</code>	<code>ae(v)</code>

<pre>if(sps_ciip_enabled_flag && sps_triangle_enabled_flag && MaxNumTriangleMergeCand > 1 && slice_type == B && cu_skip_flag[x0][y0] == 0 && (cbWidth * cbHeight) >= 64 && cbWidth < 128 && cbHeight < 128) {</pre>	
<code>ciip_flag[x0][y0]</code>	<code>ae(v)</code>

CABAC自適應

【0132】 一個實施例取決於跳過旗標(`cu_skip_flag`)添加一新上下文。若跳過旗標為0，則CABAC編碼如以前般進行。若跳過旗標為1，則使用另一格來編碼「`ciip_flag`」(參見圖18)。

【0133】 在步驟700中，檢查`cu_skip_flag`。若`cu_skip_flag`等於0，則以低值之一初始概率使用格1(步驟701)。否則，類似於沒有本文中所描述之態樣之情況，將格0與一CABAC模型(包含一初始概率)一起使用(步驟702)。接著在步驟703中應用CABAC解碼。

MRL及ISP

【0134】 此處，M表示MRL工具且N表示ISP工具。

<pre>if(sps_isp_enabled_flag && intra_luma_ref_idx[x0][y0] == 0 && (cbWidth <= MaxTbSizeY && cbHeight <= MaxTbSizeY) && (cbWidth * cbHeight > MinTbSizeY * MinTbSizeY))</pre>	
<code>intra_subpartitions_mode_flag[x0][y0]</code>	<code>ae(v)</code>

【0135】 即使MRL索引不為0，亦允許ISP模式。

CABAC自適應

【0136】 引入取決於MRL索引(`intra_luma_ref_idx`)之一新上下文來編碼`intra_subpartitions_mode_flag`。同樣，當MRL索引為0時，相同CABAC程序如以前般進行，當MRL索引不為0時，使用具有低初始概率之一新格來編碼旗標。低概率設定程序類似於前面所述之程序。

TPM及SBT

【0137】 此處，M表示TPM工具且N表示SBT工具。即使啟動TPM模式，亦允許使用SBT模式。

<code>if(CuPredMode[chType][x0][y0] == MODE_INTER && sps_sbt_enabled_flag && !ciip_flag[x0][y0] && !MergeTriangleFlag[x0][y0]) {</code>	
<code>if(cbWidth <= MaxSbtSize && cbHeight <= MaxSbtSize) {</code>	
<code>allowSbtVerH = cbWidth >= 8</code>	
<code>allowSbtVerQ = cbWidth >= 16</code>	
<code>allowSbtHorH = cbHeight >= 8</code>	
<code>allowSbtHorQ = cbHeight >= 16</code>	
<code>if(allowSbtVerH allowSbtHorH allowSbtVerQ allowSbtHorQ)</code>	
<code>cu_sbt_flag</code>	<code>ae(v)</code>
<code>}</code>	

CABAC自適應

【0138】 引入取決於TPM模式(`MergeTriangleFlag`)之一新上下文以編碼`cu_sbt_flag`。同樣，當`MergeTriangleFlag`旗標為0時，相同CABAC程序如以前般進行，當`MergeTriangleFlag`旗標不為0時，使用具有低初始概率之一新格來編碼旗標。低概率設定程序類似於前面所述之程序。

CIIP及SBT

【0139】 此處，M表示CIIP工具且N表示SBT工具。即使啟動CIIP模式，亦允許使用SBT模式。

<code>if(CuPredMode[chType][x0][y0] == MODE_INTER && sps_sbt_enabled_flag && !ciip_flag[x0][y0] && !MergeTriangleFlag[x0][y0]) {</code>	
<code>if(cbWidth <= MaxSbtSize && cbHeight <= MaxSbtSize) {</code>	
<code>allowSbtVerH = cbWidth >= 8</code>	

allowSbtVerQ = cbWidth >= 16	
allowSbtHorH = cbHeight >= 8	
allowSbtHorQ = cbHeight >= 16	
if(allowSbtVerH allowSbtHorH allowSbtVerQ allowSbtHorQ)	
cu_sbt_flag	ae(v)
}	

CABAC自適應

【0140】 引入取決於CIIP模式(`ciip_flag`)之一新上下文以編碼 `cu_sbt_flag`。當 `ciip_flag` 旗標為0時，相同CABAC程序如以前般進行，當 `ciip_flag` 旗標不為0時，使用具有低初始概率之一新格來編碼旗標。低概率設定程序類似於前面所述之程序。

LFNST(低頻率不可分離變換)及ISP

【0141】 此處，M表示ISP工具且N表示LFNST工具。即使啟動ISP模式，亦允許使用LFNST模式。

if(Min(lfnstWidth, lfnstHeight) >= 4 && sps_lfnst_enabled_flag == 1 && CuPredMode[chType][x0][y0] == MODE_INTRA && IntraSubPartitionsSplitType == ISP_NO_SPLIT && (!intra_mip_flag[x0][y0] Min(lfnstWidth, lfnstHeight) >= 16) && tu_mts_idx[x0][y0] == 0 && Max(cbWidth, cbHeight) <= MaxTbSizeY) {	
if(LfnstDcOnly == 0 && LfnstZeroOutSigCoeffFlag == 1)	
lfnst_idx[x0][y0]	ae(v)
}	

CABAC自適應

【0142】 引入取決於ISP模式(`IntraSubPartitionsSplitType`)之一新之上下文以編碼 `lfnst_idx`。當 `IntraSubPartitionsSplitType` 為 `ISP_NO_SPLIT` 時，相同CABAC程序如以前般進行，當 `IntraSubPartitionsSplitType` 不為 `ISP_NO_SPLIT` 時，使用具有低初始概率之一新格來編碼旗標。低概率設定程序類似於前面所述之程序。

LFNST僅當區塊高度或寬度小於16時才用於MIP (矩陣圖框內預測)

【0143】 此處，M表示MIP工具且N表示LFNST工具。即使啟動MIP模式且區塊大小(高度或寬度)小於16，亦允許LFNST模式。

<pre> if(Min(lfstWidth, lfstHeight) >= 4 && sps_lfst_enabled_flag == 1 && CuPredMode[chType][x0][y0] == MODE_INTRA && IntraSubPartitionsSplitType == ISP_NO_SPLIT && (!intra_mip_flag[x0][y0] Min(lfstWidth, lfstHeight) >= 16) && tu_mts_idx[x0][y0] == 0 && Max(cbWidth, cbHeight) <= MaxTbSizeY) { </pre>	
<pre> if(LfstDcOnly == 0 && LfstZeroOutSigCoeffFlag == 1) </pre>	
<pre> lfst_idx[x0][y0] </pre>	ae(v)
<pre> } </pre>	

CABAC自適應

【0144】 引入取決於MIP模式(`intra_mip_flag`)及區塊大小之一新上下文來編碼`lfst_idx`。當`intra_mip_flag`為0或寬度及高度大於16時，相同CABAC程序如以前般進行，當`intra_mip_flag`不為0且寬度及高度不大於16時，使用具有低初始概率之一新格來編碼旗標。低概率設定程序類似於前面所述之程序。

可切換濾波器

【0145】 對於經推斷模式或索引，添加一額外語法元素以顯式編碼模式或索引。

【0146】 半樣本內插濾波器索引`hpelIdx`係自AMVR (自適應運動向量解析度)模式(半像素)導出：在此情況下，一索引0或1經編碼以指示使用哪個內插濾波器。

<pre> if((sps_amvr_enabled_flag && inter_affine_flag[x0][y0] == 0 && (MvdL0[x0][y0][0] != 0 MvdL0[x0][y0][1] != 0 MvdL1[x0][y0][0] != 0 MvdL1[x0][y0][1] != 0)) (sps_affine_amvr_enabled_flag && inter_affine_flag[x0][y0] == 1 && (MvdCpL0[x0][y0][0][0] != 0 MvdCpL0[x0][y0][0][1] != 0 MvdCpL1[x0][y0][0][0] != 0 MvdCpL1[x0][y0][0][1] != 0 MvdCpL0[x0][y0][1][0] != 0 MvdCpL0[x0][y0][1][1] != 0 MvdCpL1[x0][y0][1][0] != 0 MvdCpL1[x0][y0][1][1] != 0 MvdCpL0[x0][y0][2][0] != 0 MvdCpL0[x0][y0][2][1] != 0 MvdCpL1[x0][y0][2][0] != 0 MvdCpL1[x0][y0][2][1] != 0))) { </pre>	
<pre> amvr_flag[x0][y0] </pre>	ae(v)
<pre> if(amvr_flag[x0][y0]) { </pre>	
<pre> amvr_precision_idx[x0][y0] </pre>	ae(v)
<pre> if(amvr_precision_idx[x0][y0] == HalfPelIdx) </pre>	
<pre> hpelIdx[x0][y0] </pre>	ae(v)
<pre> } </pre>	
<pre> } </pre>	

【0147】在圖框內區塊複製模式(IBC)之情況下，**hpelIdx**之值設定為等於0。在解碼程序期間，若語法元素**hpelIdx**不存在(例如，合併模式)，則將**hpelIdx**之值推斷為(**amvr_precision_idx**[x0][y0] == **HalfPelIdx**) ? IF-1 :IF-0。

【0148】在一變體中，*hpelIdx*之編碼以高級參數「**enabled_interpolation_filter_flag**」為條件。若**enabled_interpolation_filter_flag**=0，則語法元素不存在且經推斷(例如，經推斷為預設值，例如IF-0，或經推斷為「**amvr_precision_idx** [x0] [y0] == **HalfPelIdx**」)。

...	ae(v)
if(amvr_flag [x0][y0]) {	
amvr_precision_idx [x0][y0]	ae(v)
if(amvr_precision_idx [x0][y0]== HalfPelIdx && enabled_interpolation_filter_flag)	
hpelIdx [x0][y0]	ae(v)
}	
}	

【0149】在另一變體中，*hpelIdx*之編碼以高級參數「**enabled_interpolation_filter_flag**」為條件，但不以**amvr_precision_idx**之值為條件。

...	ae(v)
if(amvr_flag [x0][y0]) {	
amvr_precision_idx [x0][y0]	ae(v)
if(enabled_interpolation_filter_flag)	
hpelIdx [x0][y0]	ae(v)
}	
}	

CABAC自適應

【0150】新語法元素**hpelIdx**以一高初始概率進行CABAC編碼。高概率設定程序類似於前面所述之程序。

【0151】在一變體中，初始概率取決於

「enabled_interpolation_filter_flag」之值。

具有MRL之非MPM

【0152】 當MRL索引不為0時，允許圖框內模式為一非mpm，因此在所有情況下讀取旗標intra_luma_mpm_flag。

if(intra_luma_ref_idx[x0][y0] == 0)	
intra_luma_mpm_flag[x0][y0]	ae(v)

CABAC自適應

【0153】 引入取決於MRL索引之一新上下文來編碼intra_luma_mpm_flag。當MRL索引為0時，相同CABAC程序如以前般進行，當MRL索引不為0時，使用具有低初始概率之一新格來編碼旗標。

CIIP中之非平坦模式

【0154】 當啟動CIIP時，允許所有圖框內模式。為此，當旗標ciip_flag為真時，執行相同於圖框內模式解析之語法：解析MIP模式、MRL、MPM等。

if(sps_ciip_enabled_flag && sps_triangle_enabled_flag && MaxNumTriangleMergeCand > 1 && slice_type == B && (cbWidth * cbHeight) >= 64 && cbWidth < 128 && cbHeight < 128)	
ciip_flag[x0][y0]	ae(v)
if(ciip_flag[x0][y0]) {	
if(sps_mip_enabled_flag && (Abs(Log2(cbWidth) - Log2(cbHeight)) <= 2) && cbWidth <= MaxTbSizeY && cbHeight <= MaxTbSizeY)	
intra_mip_flag[x0][y0]	ae(v)
if(intra_mip_flag[x0][y0])	
intra_mip_mode[x0][y0]	ae(v)
else {	
if(sps_mrl_enabled_flag && ((y0 % CtbSizeY) > 0))	
intra_luma_ref_idx[x0][y0]	ae(v)
if(intra_luma_ref_idx[x0][y0] == 0)	
intra_luma_mpm_flag[x0][y0]	ae(v)
if(intra_luma_mpm_flag[x0][y0]) {	
if(intra_luma_ref_idx[x0][y0] == 0)	
intra_luma_not_planar_flag[x0][y0]	ae(v)
if(intra_luma_not_planar_flag[x0][y0])	
intra_luma_mpm_idx[x0][y0]	ae(v)
} else	

<code>intra_luma_mpm_remainder[x0][y0]</code>	<code>ae(v)</code>
<code>}</code>	
<code>}</code>	
<code>}</code>	
<code>}</code>	

CABAC自適應

【0155】 當`ciip_flag`為真時，以下所有旗標獲得一新上下文格，具有以下自適應：

- `intra_mip_flag`：初始概率設定為一低概率
- `intra_luma_ref_idx`：初始概率設定為具有索引0之一高概率
- `intra_luma_mpm_flag`：初始概率設定為一高概率
- `intra_luma_not_planar_flag`：初始概率設定為低

MMVD之合併索引

<code>if(MaxNumMergeCand > 1) {</code>	
<code>mmvd_cand_flag[x0][y0]</code>	<code>ae(v)</code>
<code>if(mmvd_cand_flag[x0][y0]) {</code>	
<code>mmvd_merge_idx_minus1[x0][y0]</code>	<code>ae(v)</code>
<code>}</code>	

【0156】 一`mmvd_merge_idx`經編碼，允許在列表中設定任何合併候選者。

【0157】 在一變體中，「`mmvd_merge_idx`」之編碼以高級參數「`enabled_mmvd_merge_idx_flag`」為條件。若`enabled_mmvd_merge_idx_flag=0`，則「`mmvd_merge_idx_minus1`」不存在且經推斷為 $(\text{mmvd_cand_flag}[x0][y0] ? 1 : 0)$ 。

<code>if(MaxNumMergeCand > 1) {</code>	
<code>mmvd_cand_flag[x0][y0]</code>	<code>ae(v)</code>
<code>if(mmvd_cand_flag[x0][y0] && enabled_mmvd_merge_idx_flag) {</code>	
<code>mmvd_merge_idx_minus1[x0][y0]</code>	<code>ae(v)</code>
<code>}</code>	

CABAC自適應

【0158】 除前2個位元係CABAC編碼之外，此自適應使用相同於

merge_idx之方法。

【0159】 第一位元以類似於原始mmvd_cand_flag之一初始概率進行CABAC編碼。第二位元以一低初始概率進行CABAC編碼(指出若索引不為0，則索引大概率為1)。與合併索引程序中一樣，剩餘位元經編碼為旁路。

【0160】 此文件描述各個態樣，包含工具、特徵、實施例、模型、方法等。此等態樣之許多者係專門描述，且至少為了展示個別特徵，通常以可聽起來可能受限之一方式來描述。然而，此係為了清楚描述，且不限制彼等態樣之應用或範疇。實際上，可將所有不同態樣組合及互換以提供進一步態樣。此外，該等態樣亦可與先前申請中所描述之態樣進行組合及互換。

【0161】 此文件中所描述及考慮之態樣可以許多不同形式實施。下文之圖12、圖13及圖19提供一些實施例，但其他實施例經考慮，且圖2、圖3及圖9之討論不限制實施方案之廣度。該等態樣之至少一者大體上係關於視訊編碼及解碼，且至少一個其他態樣大體上係關於傳輸所產生或編碼之一位元串流。此等及其他態樣可經實施為一方法、一裝置、其上儲存有用於根據所描述之方法之任何者編碼或解碼視訊資料之指令之一電腦可讀儲存媒體及/或其上儲存有根據所描述之方法之任何者產生之一位元串流之一電腦可讀儲存媒體。

【0162】 在本申請案中，術語「重建」及「解碼」可互換使用，術語「像素」及「樣本」可互換使用，術語「影像」、「圖片」及「圖框」可互換使用。通常但並非必須，術語「重建」在編碼器側使用，而「解碼」在解碼器側使用。

【0163】 本文描述各種方法，且方法之各者包括用於達成所描述之方法之一或多個步驟或動作。除非方法之適當操作需要一特定步驟或動作順序，否則可修改或組合特定步驟及/或動作之順序及/或使用。

【0164】 此文件中所描述之各種方法及其他態樣可用於修改一視訊編碼器100及解碼器200之模組，例如，圖框內預測、熵編碼及/或解碼模組(160、360、145、330)，如圖12及圖13中所展示。此外，本發明之態樣不限於VVC或HEVC，且可應用於(例如)其他標準及建議(無論預先存在或將來開發)及任何此等標準及建議(包含VVC及HEVC)之擴展。除非另有說明或技術上禁止，否則此文件中所描述之態樣可個別或組合使用。

【0165】 本文件中使用的各種數值，例如 $\{\{1,0\},\{3,1\},\{1,1\}\}$ 。特定值係出於實例目的且所描述之態樣不限於此等特定值。

【0166】 圖12繪示一編碼器100。此編碼器100之變體經考慮，但為清楚起見，下文描述編碼器100，而不描述所有預期變體。

【0167】 在經編碼之前，視訊序列可經過預編碼處理(101)，例如，對輸入色彩圖片應用一色彩變換(例如，自RGB 4:4:4至YCbCr 4:2:0之變換)，或對輸入圖片分量執行一重新映射，以便獲得對壓縮更具彈性之一信號分佈(例如，使用色彩分量之一者之一直方圖均衡化)。元資料可與預處理相關聯並附加至位元串流。

【0168】 在編碼器100中，如下文所描述，由編碼器元件編碼一圖片。以(例如) CU之單元分區(102)並處理待編碼之圖片。使用(例如)一圖框內或圖框間模式編碼各單元。當一單元以一圖框內模式編碼時，其執行圖框內預測(160)。在一圖框間模式下，執行運動估計(175)及補償(170)。編碼器決定(105)圖框內模式或圖框間模式之哪一者用於編碼單元，且藉

由(例如)一預測模式旗標來指示圖框內/圖框間決定。例如，藉由自原始影像區塊減去(110)預測區塊來計算預測殘差。

【0169】 接著變換(125)及量化(130)預測殘差。熵編碼(145)經量化之變換係數以及運動向量及其他語法元素以輸出一位元串流。編碼器可跳過變換並將量化直接應用於未變換之殘差信號。編碼器可旁通變換及量化兩者，即，殘差經直接編碼而無需應用變換或量化程序。

【0170】 編碼器解碼一經編碼區塊以為進一步預測提供一參考。反量化(140)且逆變換(150)經量化之變換係數以解碼預測殘差。在組合(155)經解碼之預測殘差及預測區塊之後，重建一影像區塊。環路濾波器(165)應用於經重建之圖片以執行(例如)去區塊/SAO (樣本自適應偏移)濾波以減少編碼偽像。經濾波影像經儲存於一參考圖片緩衝器(180)處。

【0171】 圖13繪示一視訊解碼器200之一方塊圖。在解碼器200中，如下文所描述，由解碼器元件解碼一位元串流。視訊解碼器200通常執行與如圖12中所描述之編碼遍歷互反之一解碼遍歷。編碼器100通常亦執行視訊解碼作為編碼視訊資料之部分。

【0172】 解碼器之輸入包含可由視訊編碼器100產生之一視訊位元串流。首先熵解碼(230)位元串流以獲得變換係數、運動向量及其他編碼資訊。圖片分區資訊指示圖片如何分區。因此，解碼器可根據經解碼圖片分區資訊來劃分(235)圖片。反量化(240)且逆變換(250)變換係數以解碼預測殘差。在組合(255)經解碼預測殘差及預測區塊之後，重建一影像區塊。可自圖框內預測(260)或運動補償預測(即圖框間預測)(275)獲得(270)預測區塊。環路濾波器(265)應用於經重建影像。經濾波影像儲存於一參考圖片緩衝器(280)處。

【0173】 經解碼圖片可進一步經歷後解碼處理(285)，例如，一逆色彩變換(例如，自YCbCr 4:2:0至RGB 4:4:4之變換)或一逆重新映射執行在預編碼處理(101)中執行之重新映射之逆轉。解碼後處理可使用在編碼前處理中導出並在位元串流中傳訊之元資料。

【0174】 圖19繪示其中實施各個態樣及實施例之一系統之一實施例的一方塊圖。系統1000可體現為包含以下描述之各種組件之一器件且經組態以執行此文件中所描述之態樣之一或多者。此等器件之實例包含(但不限於)各種電子器件，諸如個人電腦、膝上型電腦、智慧型電話、平板電腦、數位多媒體機頂盒、數位電視接收器、個人視訊記錄系統、經連接家用電器及伺服器。系統1000之元件可單獨或組合地體現於一單個積體電路、多個IC及/或離散組件中。例如，在至少一個實施例中，系統1000之處理及編碼器/解碼器元件跨多個IC及/或離散組件分佈。在各種實施例中，系統1000經由(例如)一通信匯流排或通過專用輸入及/或輸出埠可通信地耦合至其他類似系統或其他電子器件。在各種實施例中，系統1000經組態以實施此文件中所描述之態樣之一或多者。

【0175】 系統1000包含至少一個處理器1010，該至少一個處理器1010經組態以執行其中加載之指令用於實施(例如)此文件中所描述之各個態樣。處理器1010可包含嵌入式記憶體、輸入輸出介面及本技術中已知之各種其他電路。系統1000包含至少一個記憶體1020 (例如，一揮發性記憶體器件及/或非揮發性記憶體器件)。系統1000包含一儲存器件1040，其可包含非揮發性記憶體及/或揮發性記憶體，包含(但不限於)EEPROM、ROM、PROM、RAM、DRAM、SRAM、快閃記憶體、磁碟驅動器及/或光碟驅動器。作為非限制性實例，儲存器件1040可包含一內

部儲存器件、一經附接儲存器件及/或一網路可存取儲存器件。

【0176】 系統1000包含經組態以(例如)處理資料以提供一經編碼視訊或解碼視訊之一編碼器/解碼器模組1030，且編碼器/解碼器模組1030可包含其自身處理器及記憶體。編碼器/解碼器模組1030表示可包含於一器件中以執行編碼及/或解碼功能之模組。眾所周知，一器件可包含編碼及解碼模組之一或兩者。另外，編碼器/解碼器模組1030可實施為系統1000之一單獨元件或可作為熟習此項技術者已知之硬體及軟體之一組合併入於處理器1010內。

【0177】 可將待加載至處理器1010或編碼器/解碼器1030上以執行此文件中所描述之各個態樣之程式碼儲存於儲存器件1040中且隨後加載至記憶體1020上用於由處理器1010執行。根據各種實施例，處理器1010、記憶體1020、儲存器件1040及編碼器/解碼器模組1030之一或多者可在執行此文件中所描述之程序期間儲存各種項目之一或多者。此等經儲存項目可包含(但不限於)輸入視訊、解碼視訊或解碼視訊之部分、位元串流、矩陣、變量及來自處理等式、公式、運算及運算邏輯之中間或最終結果。

【0178】 在幾個實施例中，處理器1010及/或編碼器/解碼器模組1030內部之記憶體用於儲存指令並為在編碼或解碼期間所需之處理提供工作記憶體。然而，在其他實施例中，處理器件外部之一記憶體(例如，處理器件可為處理器1010或編碼器/解碼器模組1030)用於此等功能之一或多者。外部記憶體可為記憶體1020及/或儲存器件1040，例如，一動態揮發性記憶體及/或一非揮發性快閃記憶體。在若干實施例中，一外部非揮發性快閃記憶體用於儲存一電視之操作系統。在至少一個實施例中，一快

速、外部動態揮發性記憶體(諸如一RAM)用作用於視訊編碼及解碼操作(諸如MPEG-2、HEVC或VVC (多功能視訊編碼))之工作記憶體。

【0179】 可通過如方塊1130中所指示之各種輸入器件來提供對系統1000之元件之輸入。此等輸入器件包含(但不限於)(i)接收(例如)通過由一廣播無線傳輸之一RF信號之一RF部分，(ii)一複合輸入端子，(iii)一USB輸入端子及/或(iv)一HDMI輸入端子。

【0180】 在各種實施例中，方塊1130之輸入器件具有如本技術中已知之相關聯各自輸入處理元件。例如，RF部分可與用於以下必需之元素相關聯：(i)選擇一所要頻率(亦指稱選擇一信號，或將一信號頻帶限制於一頻帶)，(ii)降頻所選信號，(iii)再次將頻帶限制於一較窄頻帶以選擇(例如)在某些實施例中可指稱一通道之一信號頻帶，(iv)解調變降頻且頻帶受限之信號，(v)執行錯誤校正，及(vi)解多工以選擇所要資料封包流。各種實施例之RF部分包含執行此等功能之一或多個元件，例如，頻率選擇器、信號選擇器、頻帶限制器、通道選擇器、濾波器、降頻器、解調器、錯誤校正器及解多工器。RF部分可包含執行各種此等功能之一調諧器，包含(例如)將所接收之信號降頻至一較低頻率(例如，一中頻或一近基帶頻率)或基帶。在一個機頂盒實施例中，RF部分及其相關聯輸入處理元件接收在一有線(例如電纜)媒體上傳輸之一RF信號，並藉由濾波、降頻及再次濾波至一所要頻帶來執行頻率選擇。各種實施例重新配置上文所描述(及其他)元件之順序、移除此等元件之一些及/或添加執行類似或不同功能之其他元件。添加元件可包含在現有元件之間之插入元件，例如，插入放大器及一類比至數位變換器。在各種實施例中，RF部分包含一天線。

【0181】 另外，USB及/或HDMI端子可包含各自介面處理器用於跨

USB及/或HDMI連接將系統1000連接至其他電子器件。應理解，輸入處理之各個態樣(例如裡德-所羅門錯誤校正)可根據需要(例如)在一單獨輸入處理IC內或在處理器1010內實施。類似地，USB或HDMI介面處理之態樣可根據需要在單獨介面IC內或在處理器1010內實施。解調變、錯誤校正及解多工之串流經提供給各種處理元件，包含(例如)處理器1010及與記憶體及儲存元件組合操作之編碼器/解碼器1030，以根據需要處理資料串流用於在一輸出器件上呈現。

【0182】 可在一整合外殼內提供系統1000之各個元件。在整合外殼內，可使用合適連接配置1140互連各個元件並在其等之間傳輸資料，例如，本技術中已知之一內部匯流排，包含I2C匯流排、佈線及印刷電路板。

【0183】 系統1000包含使得能夠經由通信通道1060與其他器件通信之通信介面1050。通信介面1050可包含(但不限於)經組態以通過通信通道1060傳輸及接收資料之一收發器。通信介面1050可包含(但不限於)一數據機或網卡，且通信通道1060可(例如)在一有線及/或無線媒體內實施。

【0184】 在各種實施例中，使用一無線網路(諸如IEEE 802.11)將資料串流至系統1000。此等實施例之無線信號在(例如)適於Wi-Fi通信之通信通道1060及通信介面1050上方經接收。此等實施例之通信通道1060通常連接至一存取點或路由器，該存取點或路由器提供對包含網際網路之外部網路之存取用於允許串流應用程式及其他空中通信。其他實施例使用一機頂盒向系統1000提供串流資料，該機頂盒通過輸入區塊1130之HDMI連接傳遞資料。其他實施例亦使用輸入區塊1130之RF連接向系統1000提供串流資料。

【0185】 系統1000可向包含一顯示器1100、揚聲器1110及其他周邊器件1120之各種輸出器件提供一輸出信號。在實施例之各種實例中，其他周邊器件1120包含一獨立DVR、一碟播放機、一立體聲系統、一照明系統及基於系統1000之輸出提供一功能之其他器件的一或多者。在各種實施例中，控制信號在系統1000與顯示器1100、揚聲器1110或其他周邊器件1120之間使用傳訊(諸如AV.Link、CEC或在有或無使用者干預之情況下實現器件至器件之控制的其他通信協定)通信。輸出器件可經由通過各自介面1070、1080及1090之專用連接通信地耦合至系統1000。替代地，輸出器件可經由通信介面1050使用通信通道1060連接至系統1000。顯示器1100及揚聲器1110可與一電子器件(例如一電視)中之系統1000之其他組件整合於一單一單元中。在各種實施例中，顯示介面1070包含一顯示驅動器，例如，一時序控制器(T Con)晶片。

【0186】 例如，若輸入1130之RF部分係一單獨機頂盒之部分，則顯示器1100及揚聲器1110可替代地與其他組件之一或多者分開。在其中顯示器1100及揚聲器1110係外部組件之各種實施例中，可經由包含(例如)HDMI埠、USB埠或COMP輸出之專用輸出連接來提供輸出信號。

【0187】 實施例可藉由由處理器1010實施之電腦軟體或藉由硬體、或藉由硬體及軟體之一組合來執行。作為一非限制性實例，實施例可由一或多個積體電路實施。記憶體1020可為適於技術環境之任何類型且可使用任何適當資料儲存技術(諸如作為非限制性實例之光學記憶體器件、磁記憶體器件、基於半導體之記憶體器件、固定記憶體及可移動記憶體器)實施。處理器1010可為適於技術環境之任何類型，且可包含作為非限制性實例之微處理器、通用電腦、專用電腦及基於一多核心架構之處理器之

一或多者。

【0188】 各種實施方案涉及解碼。如本申請案中所使用，「解碼」可涵蓋(例如)對一所接收之編碼序列執行以產生適於顯示之一最終輸出之全部或部分處理。在各種實施例中，此等程序包含通常由一解碼器執行之程序之一或多者，例如，熵解碼、逆量化、逆變換及差分解碼。在各種實施例中，此等處理亦或替代地包含由本申請案中所描述之各種實施方案之一解碼器執行之處理，例如，提取待用於各種圖框內預測參考陣列之權重之一索引。

【0189】 作為進一步實例，在一個實施例中，「解碼」僅係指熵解碼，在另一實施例中，「解碼」僅係指差分解碼，且在另一實施例中，「解碼」係指熵解碼及差分解碼之一組合。基於特定描述之上下文，片語「解碼程序」旨在專門係指操作之一子集或大體上係指更廣泛解碼程序將係清楚的，且據信熟習此項技術者很好理解。

【0190】 各種實施方案涉及編碼。依與上文關於「解碼」之討論類似之一方式，如在本申請案中使用之「編碼」可涵蓋(例如)對一輸入視訊序列執行以產生一編碼位元串流之全部或部分處理。在各種實施例中，此等程序包含通常由一編碼器執行之程序之一或多者，例如，分區、差分編碼、變換、量化及熵編碼。在各種實施例中，此等程序亦或替代地包含由本申請案中所描述之各種實施方案之一編碼器執行之程序，例如，圖框內預測參考陣列之加權。

【0191】 作為進一步實例，在一個實施例中，「編碼」僅係指熵編碼，在另一實施例中，「編碼」僅係指差分編碼，且在另一實施例中，「編碼」係指差分編碼及熵編碼之一組合。基於特定描述之上下文，片語「編

碼程序」旨在專門係指操作之一子集或大體上係指更廣泛編碼程序將係清楚的，且據信熟習此項技術者很好理解。

【0192】 注意，如本文中所使用之語法元素係描述性術語。因而，其等不排除使用其他語法元素名稱。

【0193】 當將一圖作為一流程圖呈現時，應理解，其亦提供一對應裝置之一方塊圖。類似地，當將一圖作為一方塊圖呈現時，應理解，其亦提供一對應方法/程序之一流程圖。

【0194】 各種實施例係指速率失真計算或速率失真最佳化。在編碼程序期間，通常給予計算複雜性之約束，通常考慮速率與失真之間的平衡或折衷。通常將速率失真最佳化公式化為最小化一速率失真函數，該函數係速率及失真之一加權總和。存在多種解決速率失真最佳化問題之方法。例如，該等方法可基於對所有編碼選項之一廣泛測試，包含所有考慮之模式或編碼參數值，對其編碼成本及編碼及解碼之後之重建信號之相關失真之一完整評估。亦可使用更快方法來節省編碼複雜性，特別係基於預測或預測殘差信號而非重建信號來計算一近似失真。亦可使用此等兩種方法之混合，諸如僅對一些可行編碼選項使用一近似失真，且對其他編碼選項使用一完整失真。其他方法僅評估可行編碼選項之一子集。更一般而言，許多方法採用多種技術之任何者來執行最佳化，但最佳化並不一定係編碼成本及相關失真兩者之一完整評估。

【0195】 本文中所描述之實施方案及態樣可(例如)以一方法或一程序、一裝置、一軟體程式、一資料串流或一信號實施。即使僅在呈一單一形式之實施方案之上下文中進行討論(例如，僅作為一方法討論)，所討論之特徵之實施方案亦可呈其他形式(例如，一裝置或程式)實施。一裝置可

(例如)以適當硬體、軟體及韌體實施。該等方法可(例如)在一處理器中實施，該處理器通常係指處理器件，包含(例如)一電腦、一微處理器、一積體電路或一可程式化邏輯器件。處理器亦包含通信器件，諸如(例如)電腦、手機、可攜式/個人數位助理(「PDA」)及促進終端使用者之間的資訊通信之其他器件。

【0196】 提及「一個實施例」或「一實施例」或「一個實施方案」或「一實施方案」以及其等之其他變體意謂結合實施例描述之一特定特徵、結構、特性等包含於至少一個實施例中。因此，在貫穿此文件之各處出現之片語「在一個實施例中」或「在一實施例中」或「在一種實施方案中」或「在一種實施方案中」以及任何其他變體之出現並不一定全部指代相同實施例。

【0197】 另外，此文件可係指「判定」各種資訊。判定資訊可包含(例如)估計資訊、計算資訊、預測資訊或自記憶體擷取資訊之一或多者。

【0198】 進一步言之，此文件可係指「存取」各種資訊。存取資訊可包含(例如)接收資訊、擷取資訊(例如自記憶體)、儲存資訊、移動資訊、複製資訊、計算資訊、判定資訊、預測資訊或估計資訊之一或多者。

【0199】 另外，此文件可係指「接收」各種資訊。與「存取」一樣，接收旨在一廣義術語。接收資訊可包含(例如)存取資訊或(例如自記憶體)擷取資訊之一或多者。進一步言之，在操作(諸如，例如，儲存資訊、處理資訊、傳輸資訊、移動資訊、複製資訊、擦除資訊、計算資訊、判定資訊、預測資訊或估計資訊)期間，通常依一種或另一種方式涉及「接收」。

【0200】 應瞭解，例如在「A/B」、「A及/或B」及「A及B之至少一

者」之情況下使用以下「/」、「及/或」及「至少一者」之任何者旨在涵蓋僅選擇第一所列選項(A)，或僅選擇第二所列選項(B)，或選擇兩個選項(A及B)。作為一進一步實例，在「A、B及/或C」及「A、B及C之至少一者」之情況下，此片語旨在涵蓋僅選擇第一所列選項(A)，或僅選擇第二所列選項(B)或僅選擇第三所列選項(C)，或僅選擇第一及第二所列選項(A及B)，或僅選擇第一及第三所列選項(A及C)，或僅選擇第二及第三所列選項(B及C)，或選擇所有三個選項(A及B及C)。如本技術及相關技術之一般技術者清楚般，對於所列之盡可能多之項目，此可擴展。

【0201】 同樣地，如本文中所使用，字詞「信號」尤其係指向一對應解碼器指示一些東西。例如，在某些實施例中，編碼器傳訊待用於圖框內預測參考陣列之複數個權重之一特定權重。依此方式，在一實施例中，在編碼器側及解碼器側兩者處使用相同參數。因此，例如，一編碼器可向解碼器傳輸(顯式傳訊)一特定參數，使得解碼器可使用相同特定參數。相反，若解碼器已具有特定參數以及其他參數，則可使用傳訊而不傳輸(隱式傳訊)以簡單地允許解碼器知道並選擇特定參數。藉由避免傳輸任何實際功能，在各種實施例中實現一位元節省。應瞭解，可依多種方式完成傳訊。例如，在各種實施例中，一或多個語法元素、旗標等用於將資訊傳訊給一對應解碼器。儘管前面涉及字詞「信號」之動詞形式，但字詞「信號」在本文中亦可用作一名詞。

【0202】 如一般技術者將明白，實施方案可產生各種信號，該等信號經格式化以攜帶可(例如)經儲存或傳輸之資訊。該資訊可包含(例如)用於執行一方法之指令或由所描述之實施方案之一者產生之資料。例如，一信號可經格式化以攜帶一所描述之實施例之位元串流。可將此一信號格式

化為(例如)一電磁波(例如，使用頻譜之一射頻部分)或一基帶信號。格式化可包含(例如)編碼一資料串流且利用經編碼資料串流來調變一載波。信號攜帶之資訊可為(例如)類比或數位資訊。眾所周知，信號可通過各種不同有線或無線鏈路傳輸。信號可儲存於一處理器可讀媒體上。

【0203】 實施例可單獨地或組合地包含跨各種不同申請專利範圍類別及類型之以下特徵或實體之一或多者：

- 放寬一視訊編碼標準中之硬約束以改良編碼模式
- 通過語法改變實施上文硬約束之放寬
- 通過特設熵上下文實施上文硬約束之放寬
- 通過改變熵初始概率參數實施上文硬約束之放寬。
- 一編碼/解碼方案，其中添加指示是否啟動一編碼模式之一CABAC上下文。
- 一模式推斷模式，其中移除模式排除及/或使用一編碼參數之一條件來控制/啟動經推斷值態樣(用於顯式編碼值之新語法元素)
- 一可切換濾波器模式，其中 *hpellIfIdx* 之編碼以高級參數「enabled_interpolation_filter_flag」為條件。
- 一模式，其中編碼 *mmvd_merge_idx*，允許在列表中設定任何合併候選者。
- 在一變體中，「**mmvd_merge_idx**」之編碼以高級參數「enabled_mmvd_merge_idx_flag」為條件。
- 當使用一模式N為一組初始區塊啟動一模式M時，計算模式N之一偏差率失真成本。
- 取決於一跳過旗標添加一新上下文。

- 包含所描述語法元素或其變體之一或多者之一位元串流或信號。
- 創建及/或傳輸及/或接收及/或解碼包含所描述語法元素或其變體之一或多者之一位元串流或信號。
- 根據所描述之實施例之任何者執行環路濾波之一電視、機頂盒、手機、平板電腦或其他電子器件。
- 根據所描述之實施例之任何者執行環路過濾且顯示(例如，使用一監測器、螢幕或其他類型之顯示器)一結果影像之一電視、機頂盒、手機、平板電腦或其他電子器件。
- 調諧(例如，使用一調諧器)一頻道以接收包含一經編碼影像之一信號且根據所描述之實施例之任何者執行環路濾波之一電視、機頂盒、手機、平板電腦或其他電子器件。
- 通過空中接收(例如，使用一天線)包含一經編碼影像之一信號且根據所描述之實施例之任何者執行環路濾波之一電視、機頂盒、手機、平板電腦或其他電子器件。

【0204】 在圖20中展示根據此處所描述之一般態樣之一方法2000之一個實施例。方法在開始方塊2001處開始，且控制進行至方塊2010用於判定是否啟動一第一編碼工具用於編碼一視訊影像之一部分。控制自方塊2010進行至方塊2020用於判定是否一第二編碼工具經啟動用於編碼視訊影像之部分及用於該啟動之條件。控制自方塊2020進行至方塊2030用於有條件地執行該等編碼工具之至少一者。

【0205】 在圖21中展示根據此處所描述之一般態樣之一方法2100之一個實施例。該方法在開始方塊2101處開始且控制進行至方塊2110用於判定是否自一視訊位元串流啟動一第一解碼工具。控制自方塊2110進行至

方塊2120用於判定是否一第二解碼工具經啟動及用於該啟動之條件。控制自方塊2120進行至方塊2130用於有條件地執行該等解碼工具之至少一者。

【0206】 圖22展示一裝置2200之一個實施例，該裝置2200基於基於鄰近樣本之參數模型，使用簡化編碼模式來編碼、解碼、壓縮或解壓縮視訊資料。該裝置包括處理器2210且可通過至少一個埠互連至一記憶體2220。處理器2210及記憶體2220兩者亦可具有至外部連接之一或多個額外互連。

【0207】 處理器2210亦經組態以使用所描述態樣之任何者在一位元串流中插入或接收資訊，以及壓縮、編碼或解碼。

【0208】 貫穿本說明書，亦支持及考慮各種其他廣義以及特化之發明及申請專利範圍。

【符號說明】

【0209】

100:視訊編碼器

102:分區

105:決定

110:減去

125:變換

130:量化

140:反量化

145:熵編碼

150:逆變換

155:組合
160:圖框內預測
165:環路濾波器
170:補償
175:運動估計
180:參考圖片緩衝器
200:解碼器
220:熵解碼
240:反量化
250:逆變換
255:組合
260:圖框內預測
265:環路濾波器
270:獲得
275:運動補償
280:參考圖片緩衝器
300:步驟
301:步驟
302:步驟
303:步驟
304:步驟
305:步驟
400:步驟

401:步驟

402:步驟

403:步驟

404:步驟

405:步驟

406:步驟

500:步驟

501:步驟

502:步驟

503:步驟

504:步驟

505:步驟

600:步驟

601:步驟

602:步驟

603:步驟

604:步驟

605:步驟

606:步驟

700:步驟

701:步驟

702:步驟

703:步驟

1000:系統
1010:處理器
1020:記憶體
1030:編碼器/解碼器模組
1040:儲存器件
1050:通信介面
1060:通信通道
1070:顯示介面
1080:介面
1090:介面
1100:顯示器
1110:揚聲器
1120:周邊器件
1130:輸入區塊
1140:連接配置
2000:方法
2001:開始方塊
2010:方塊
2020:方塊
2030:方塊
2100:方法
2101:開始方塊
2110:方塊

2120:方塊

2130:方塊

2200:裝置

2210:處理器

2220:記憶體

【發明申請專利範圍】

【請求項1】

一種用於視訊編碼之方法，其包括：

判定是否啟動用於編碼一視訊影像之一部分之一第一編碼工具；

判定是否一第二編碼工具經啟動用於編碼該視訊影像之該部分；

回應於該等判定執行該等編碼工具之至少一者；

編碼一第一語法元素(syntax element)，其指示該第一編碼工具是否經啟動；及

編碼一第二語法元素，其指示該第二編碼工具是否經啟動，該第二語法元素係使用至少基於該第一語法元素導出之一上下文格(context bin)來編碼。

【請求項2】

如請求項1之方法，其中當該第一編碼工具經啟動時所使用的該上下文格具有一低概率被初始化。

【請求項3】

如請求項1之方法，其中當該第一編碼工具經啟動時所導出的該上下文格不同於當該第一編碼工具未經啟動時所使用的該上下文格。

【請求項4】

如請求項1之方法，其中該第一編碼工具係具有區塊大小(block size)小於一值之矩陣圖框內預測(matrix intra prediction)，且該第二編碼工具係低頻率不可分離變換(Low Frequency Non-Separable Transform)。

【請求項5】

如請求項1之方法，其中該第一編碼工具係跳過模式，且該第二編碼

工具係經組合圖框內-圖框間預測(Combined Intra-Inter Prediction)。

【請求項6】

如請求項1之方法，其中該第一編碼工具係多個參考線，且該第二編碼工具係圖框內子區塊分區(intra sub-block partitioning)。

【請求項7】

一種用於視訊編碼之裝置，其包括：

一處理器，其經組態以：

判定是否啟動用於編碼一視訊影像之一部分之一第一編碼工具；

判定是否一第二編碼工具經啟動用於編碼該視訊影像之該部分；

回應於該等判定執行該等編碼工具之至少一者；

編碼一第一語法元素，其指示該第一編碼工具是否經啟動；及

編碼一第二語法元素，其指示該第二編碼工具是否經啟動，該第二語法元素係使用至少基於該第一語法元素導出之一上下文格來編碼。

【請求項8】

如請求項7之裝置，其中當該第一編碼工具經啟動時所使用的該上下文格具有一低概率被初始化。

【請求項9】

如請求項7之裝置，其中當該第一編碼工具經啟動時所導出的該上下文格不同於當該第一編碼工具未經啟動時所使用的該上下文格。

【請求項10】

如請求項7之裝置，其中該第一編碼工具係具有區塊大小小於一值之矩陣圖框內預測，且該第二編碼工具係低頻率不可分離變換。

【請求項11】

如請求項7之裝置，其中該第一編碼工具係跳過模式，且該第二編碼工具係經組合圖框內-圖框間預測。

【請求項12】

如請求項7之裝置，其中該第一編碼工具係多個參考線，且該第二編碼工具係圖框內子區塊分區。

【請求項13】

一種用於視訊解碼之方法，其包括：

解碼一第一語法元素，其指示一第一解碼工具是否經啟動；

解碼一第二語法元素，其指示一第二解碼工具是否經啟動，該第二語法元素係使用至少基於該第一語法元素導出之一上下文格來解碼；及

基於該經解碼第一語法元素及該經解碼第二語法元素執行該等解碼工具之至少一者。

【請求項14】

如請求項13之方法，其中當該第一解碼工具經啟動時所使用的該上下文格具有一低概率被初始化。

【請求項15】

如請求項13之方法，其中當該第一解碼工具經啟動時所導出的該上下文格不同於當該第一解碼工具未經啟動時所使用的該上下文格。

【請求項16】

如請求項13之方法，其中該第一解碼工具係具有區塊大小小於一值之矩陣圖框內預測，且該第二解碼工具係低頻率不可分離變換。

【請求項17】

如請求項13之方法，其中該第一解碼工具係跳過模式，且該第二解碼工具係經組合圖框內-圖框間預測。

【請求項18】

如請求項13之方法，其中該第一解碼工具係多個參考線，且該第二解碼工具係圖框內子區塊分區。

【請求項19】

一種用於視訊解碼之裝置，其包括：

一處理器，其經組態以：

解碼一第一語法元素，其指示一第一解碼工具是否經啟動；

解碼一第二語法元素，其指示一第二解碼工具是否經啟動，該第二語法元素係使用至少基於該第一語法元素導出之一上下文格解碼；及

基於該經解碼第一語法元素及該經解碼第二語法元素執行該等解碼工具之至少一者。

【請求項20】

如請求項19之裝置，其中當該第一解碼工具經啟動時所使用的該上下文格具有一低概率被初始化。

【請求項21】

如請求項19之裝置，其中當該第一解碼工具經啟動時所導出的該上下文格不同於當該第一解碼工具未經啟動時所使用的該上下文格。

【請求項22】

如請求項19之裝置，其中該第一解碼工具係具有區塊大小小於一值之矩陣圖框內預測，且該第二解碼工具係低頻率不可分離變換。

【請求項23】

如請求項19之裝置，其中該第一解碼工具係跳過模式，且該第二解碼工具係經組合圖框內-圖框間預測。

【請求項24】

如請求項19之裝置，其中該第一解碼工具係多個參考線，且該第二解碼工具係圖框內子區塊分區。

【請求項25】

一種用於視訊解碼之器件，其包括：

如請求項19之一裝置；及

以下至少一者：(i)一天線，其經組態以接收一信號，該信號包含一視訊區塊；(ii)一頻帶限制器，其經組態以將該所接收之信號限制至包含該視訊區塊之一頻率帶；及(iii)一顯示器，其經組態以顯示代表該視訊區塊之一輸出。

【請求項26】

一種非暫時性電腦可讀媒體，其含有如請求項1之方法產生之資料內容用於使用一處理器回放。

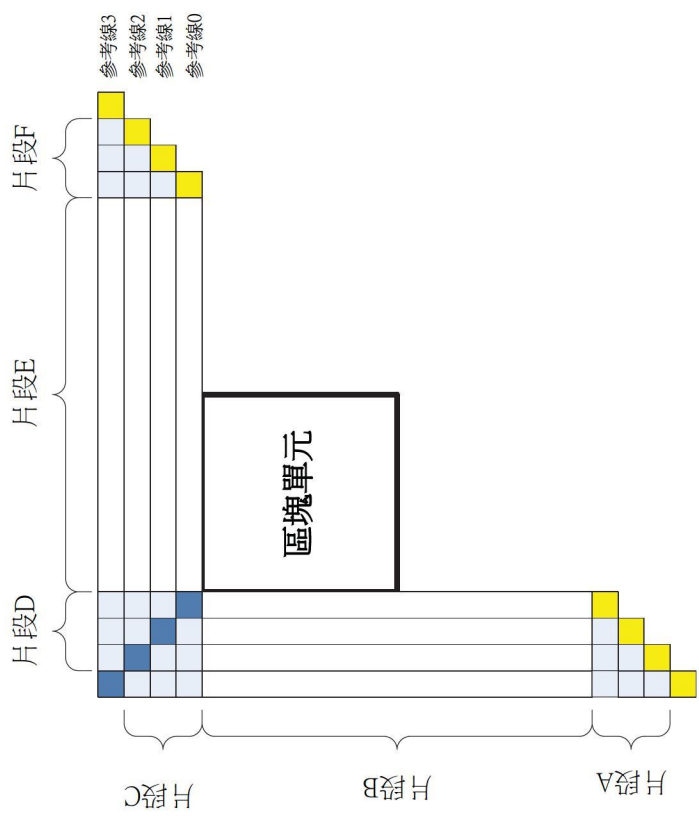
【請求項27】

一種非暫時性電腦程式產品，其包括經儲存指令，當該程式由一電腦執行時，該等指令致使該電腦執行如請求項13之方法。

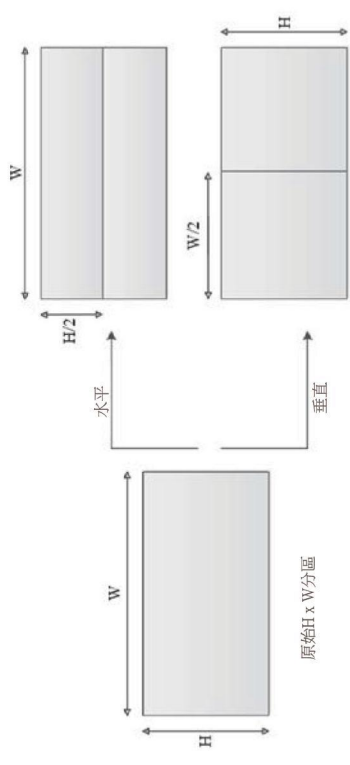
【請求項28】

一種非暫時性電腦程式產品，其包括經儲存指令，當該程式由一電腦執行時，該等指令致使該電腦執行如請求項1之方法。

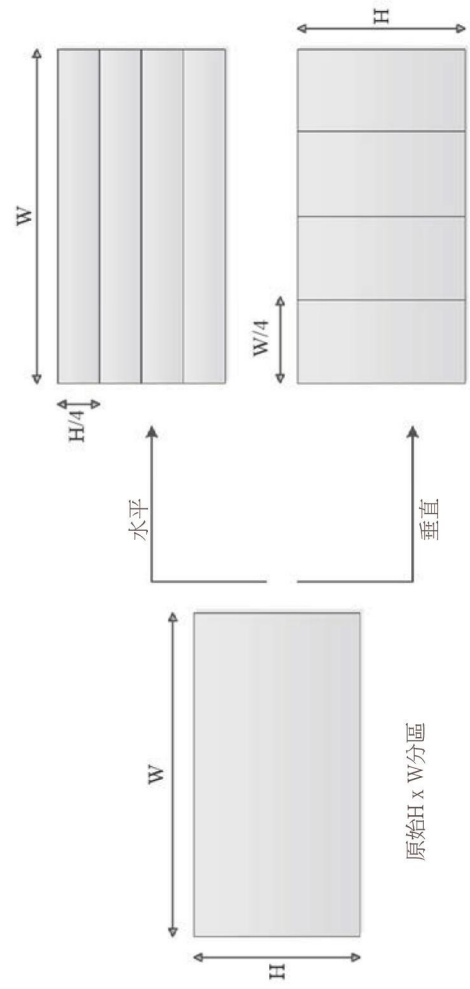
【發明圖式】



【圖1】

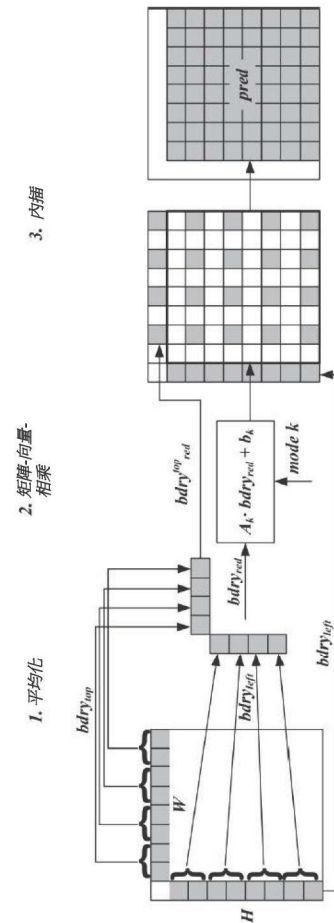


a) 4 x 8及8 x 4 CU之子分區之實例

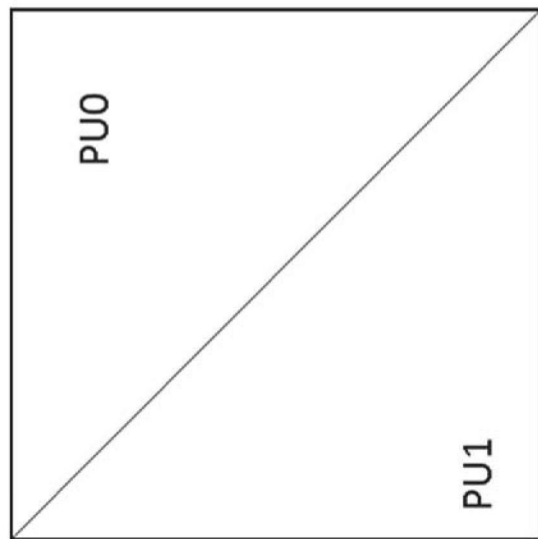
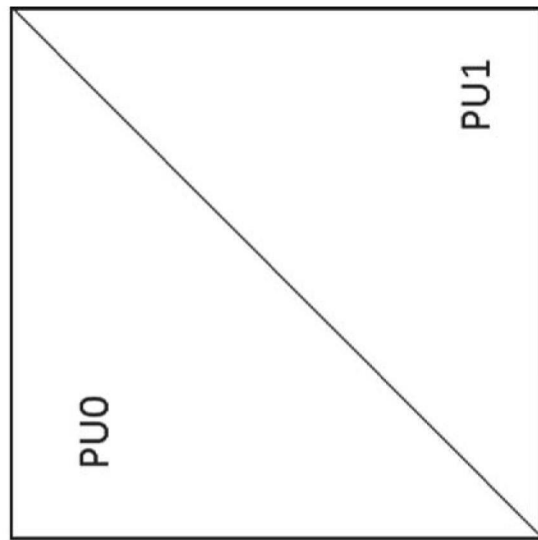


b) 除4 x 8及8 x 4及4 x 4之外之CU之子分區之實例

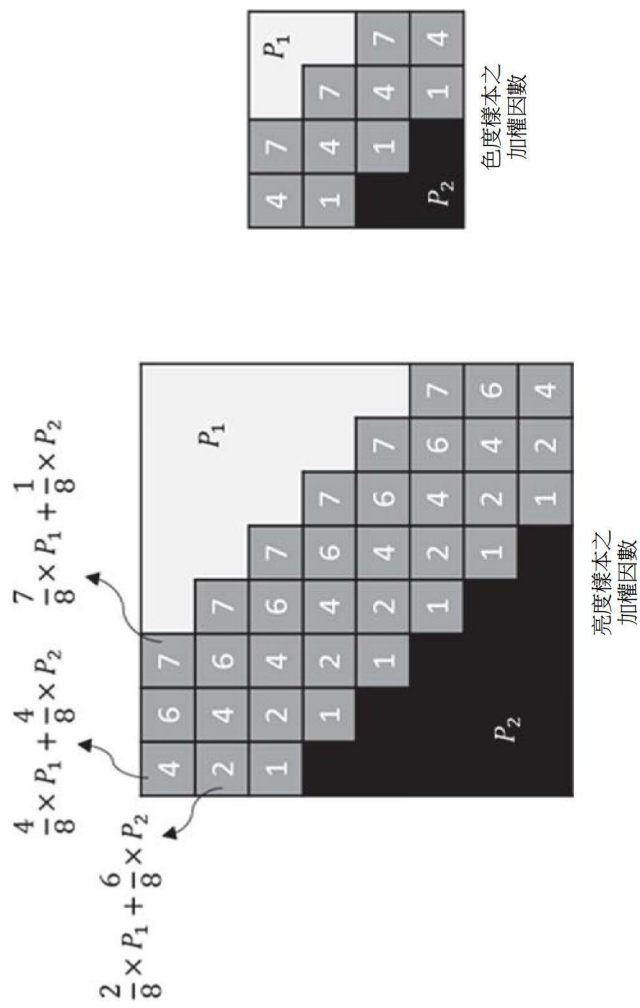
【圖2】



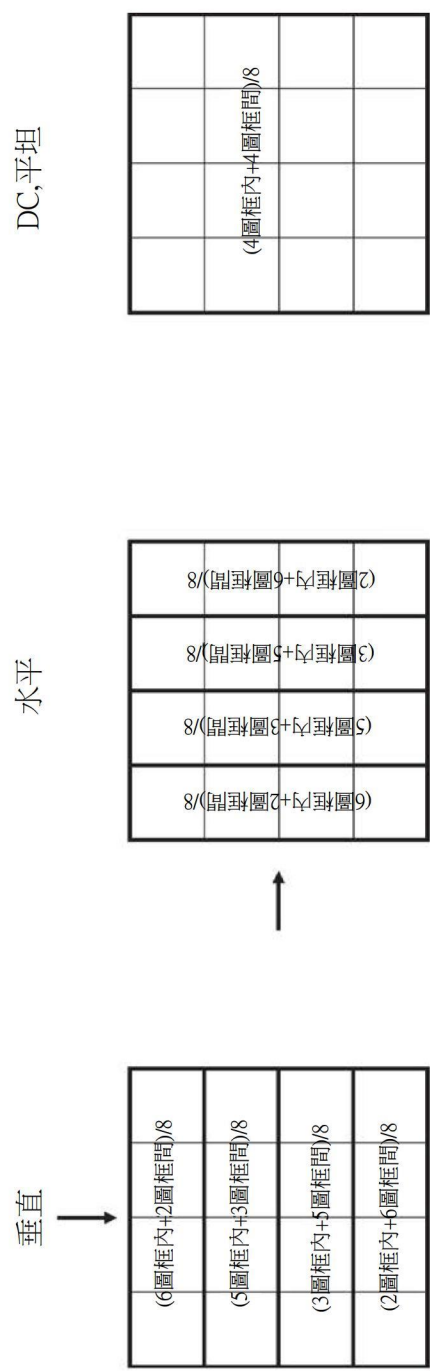
【圖3】



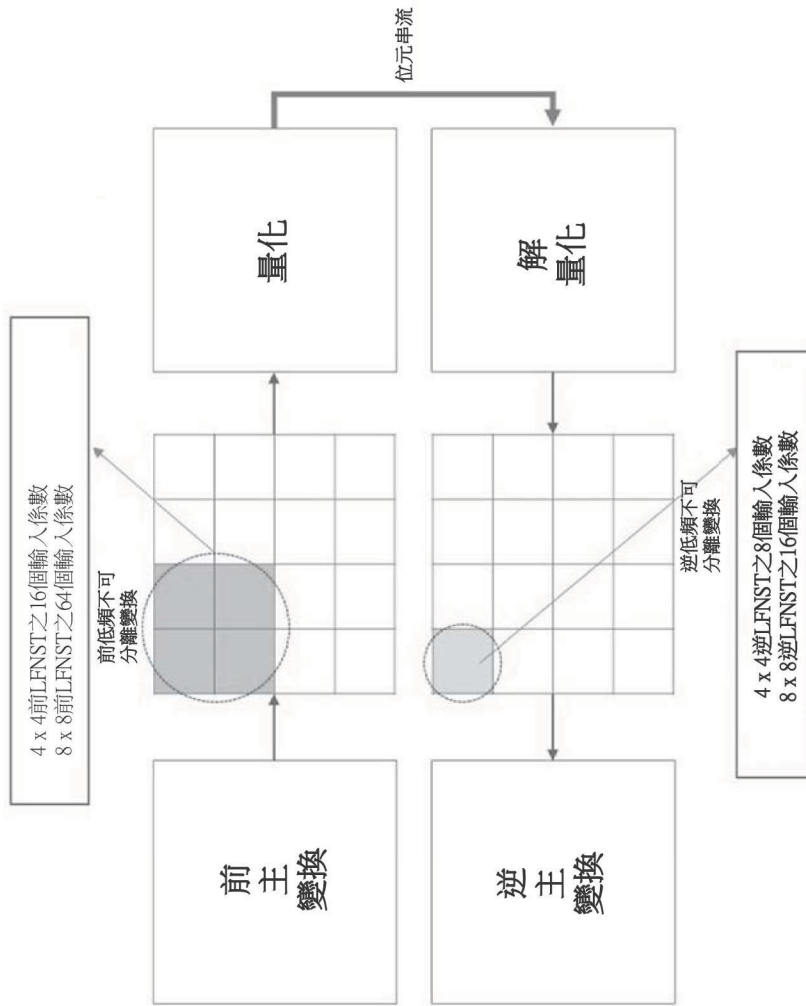
【圖4】



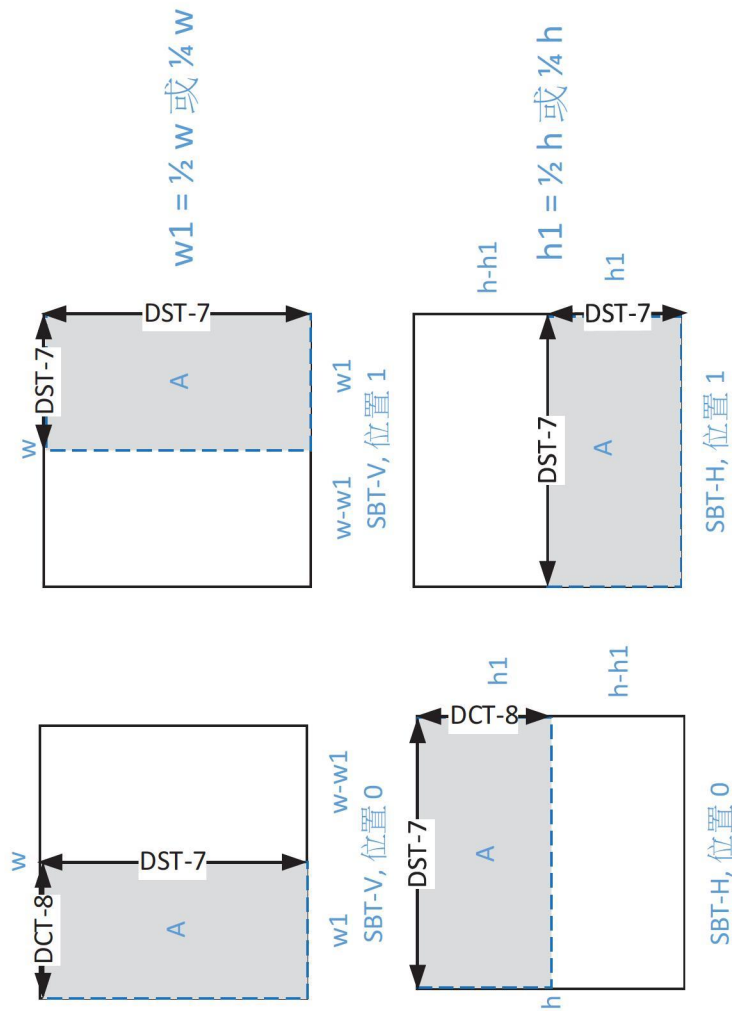
【圖5】



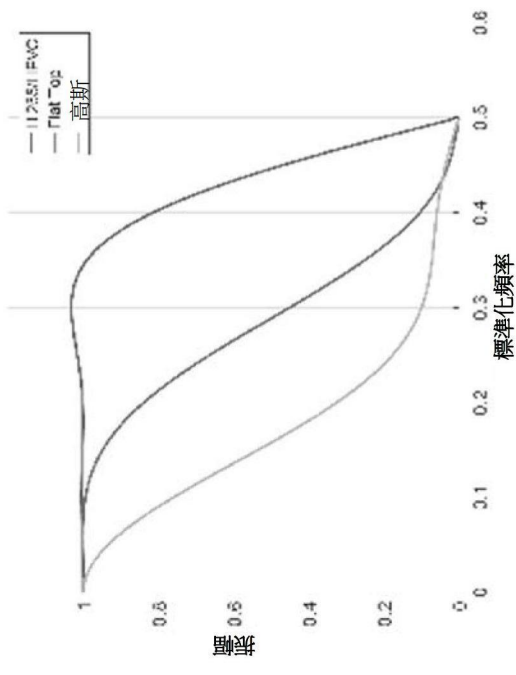
【圖6】



【圖7】

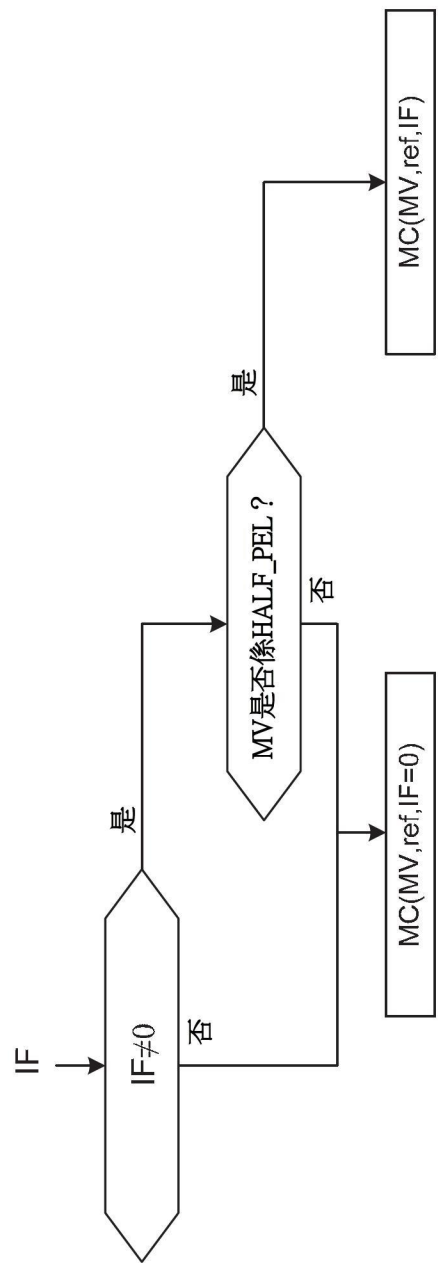


【圖8】



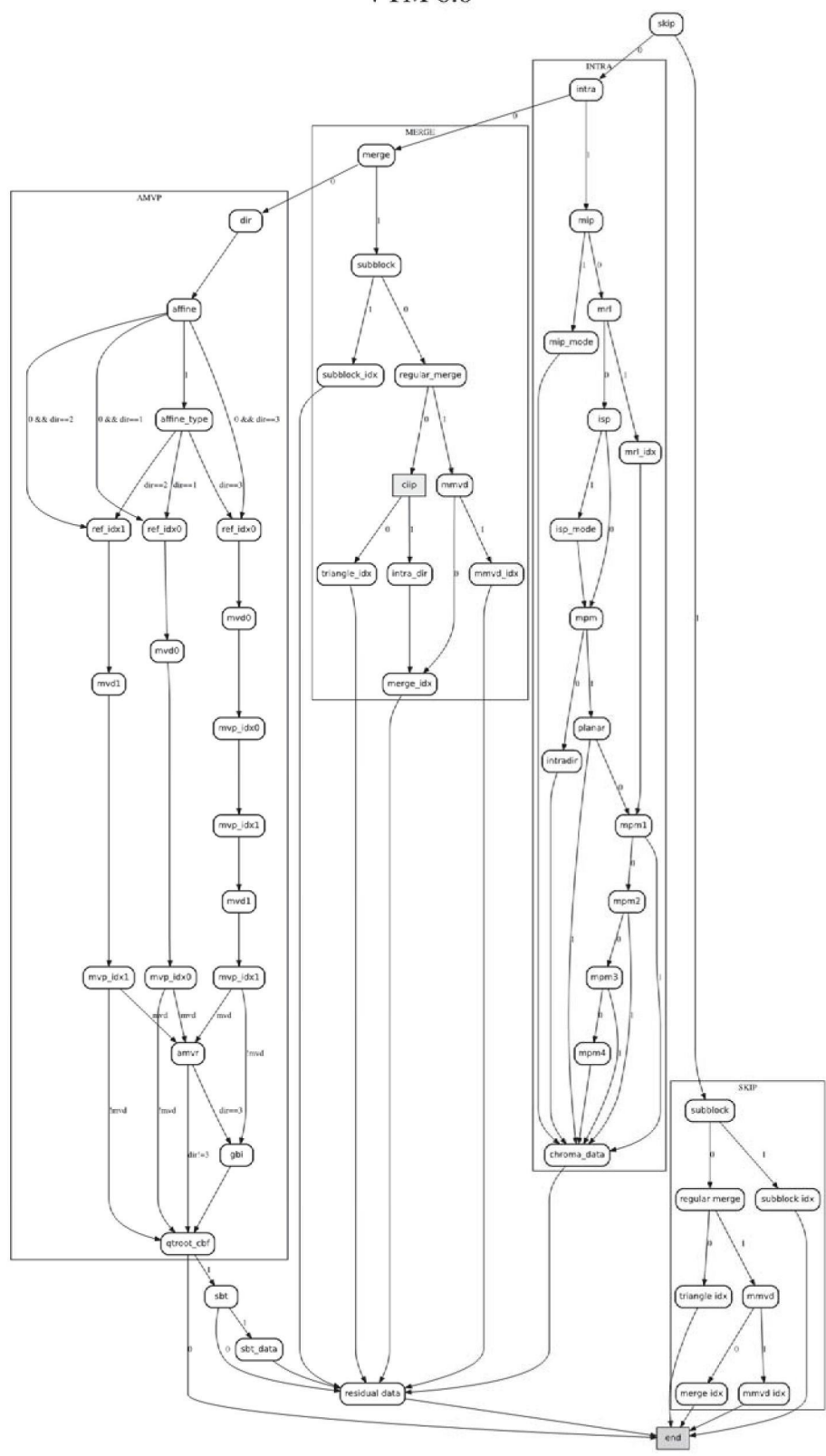
濾波器	係數 $[i]$
HEVC	$[-1 \ 4 \ -11 \ 40 \ 40 \ -11 \ 4 \ -1]$
平頂	$[-3 \ 4 \ 31 \ 31 \ 4 \ -3]$
高斯	$[3 \ 9 \ 20 \ 20 \ 9 \ 3]$

【圖9】

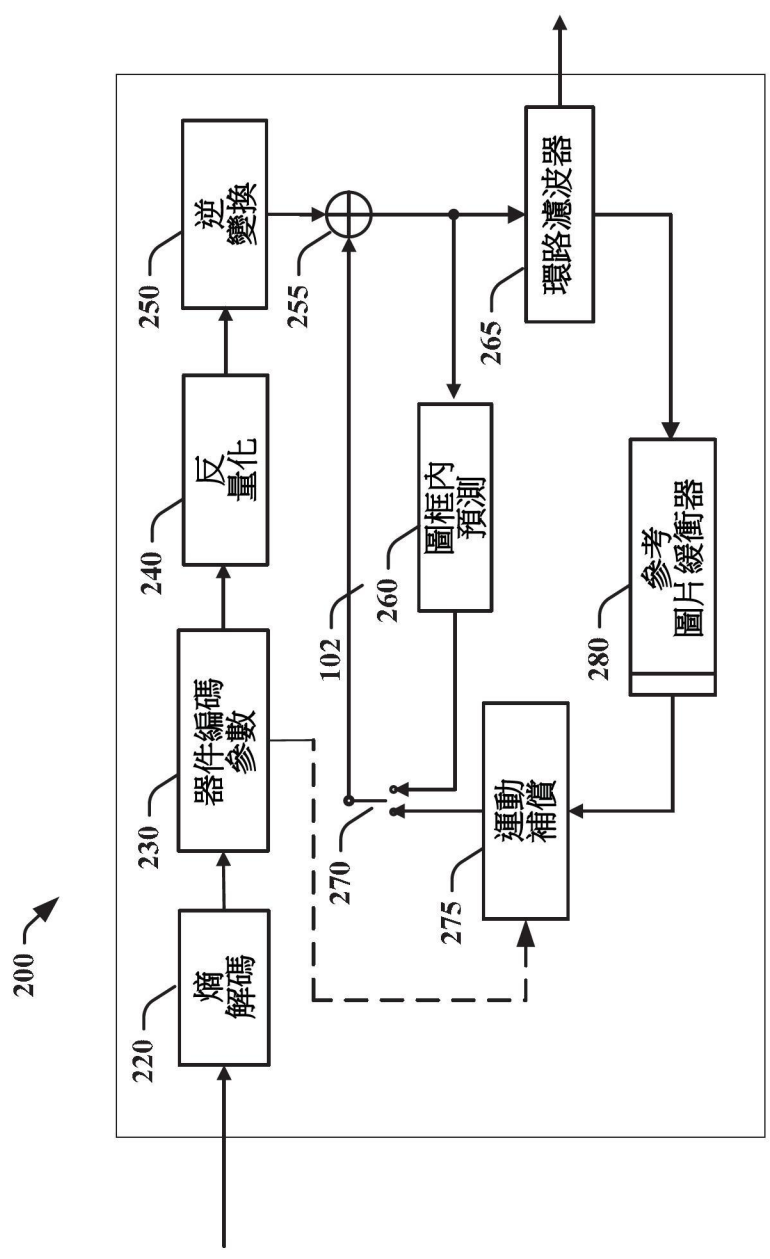


【圖10】

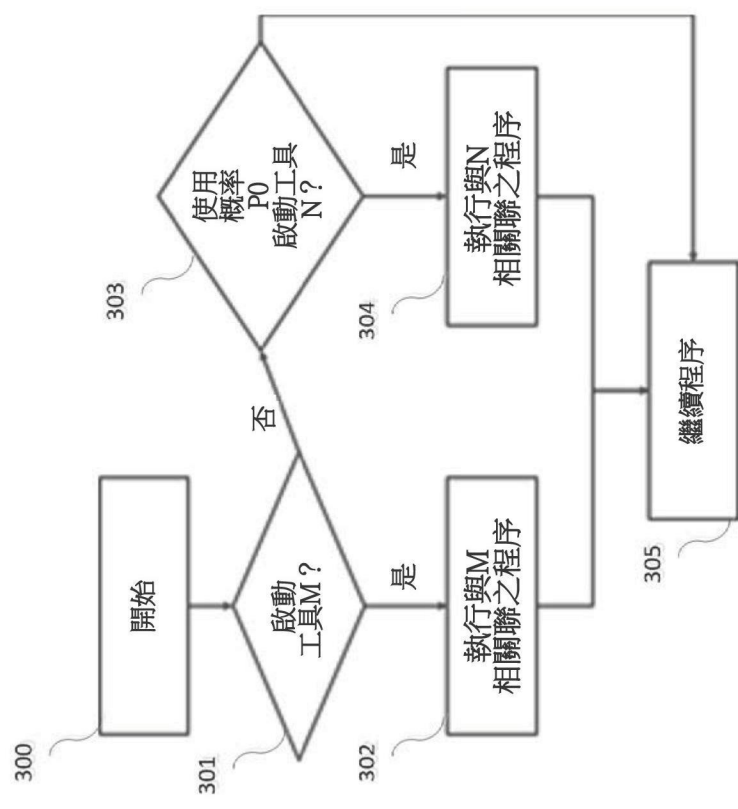
VTM 6.0



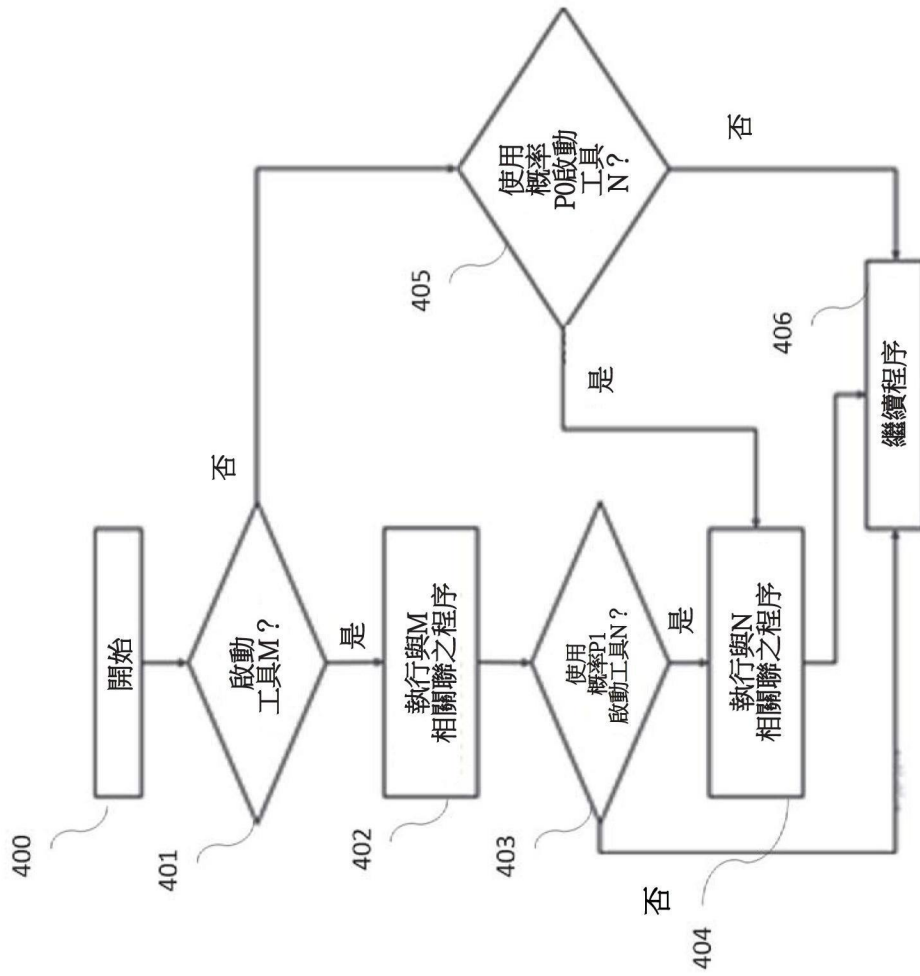
【圖11】



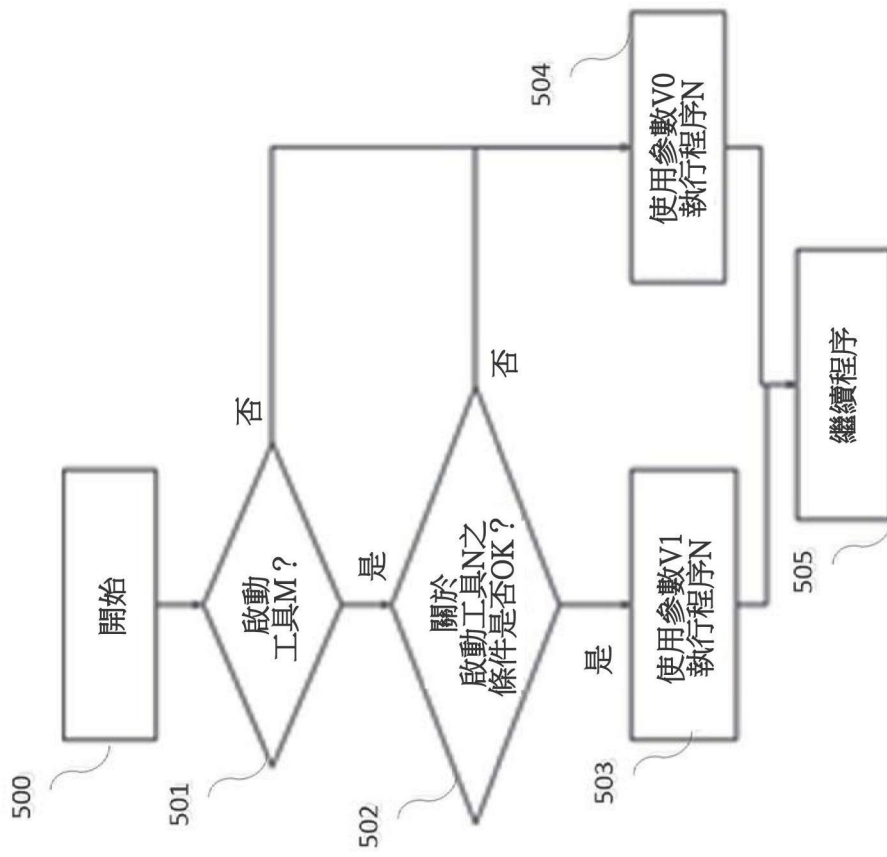
【圖13】



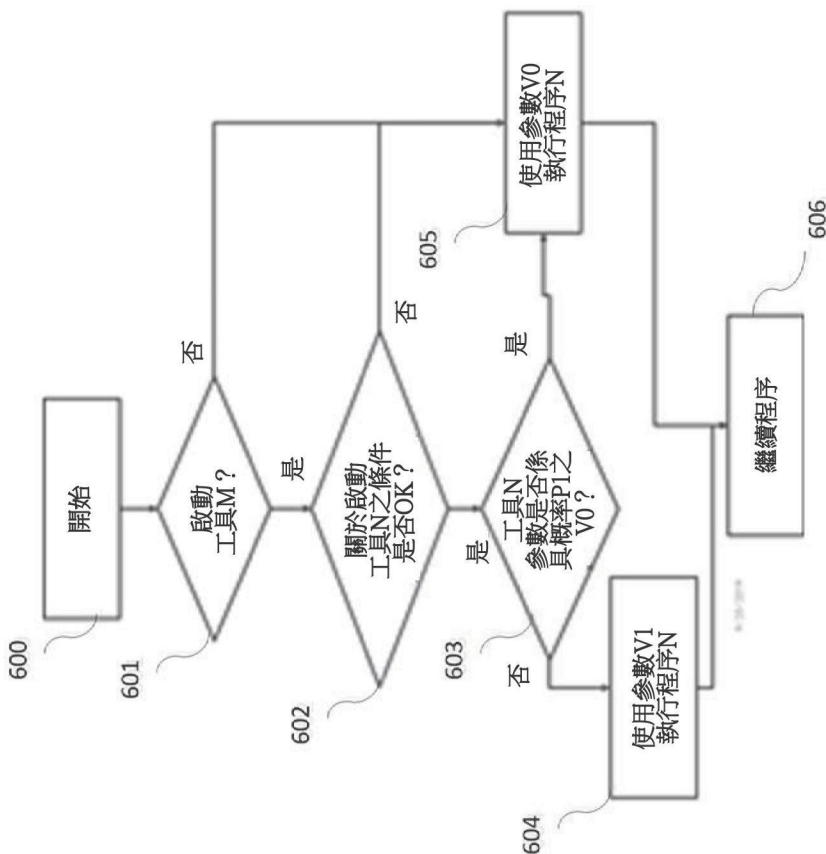
【圖14】



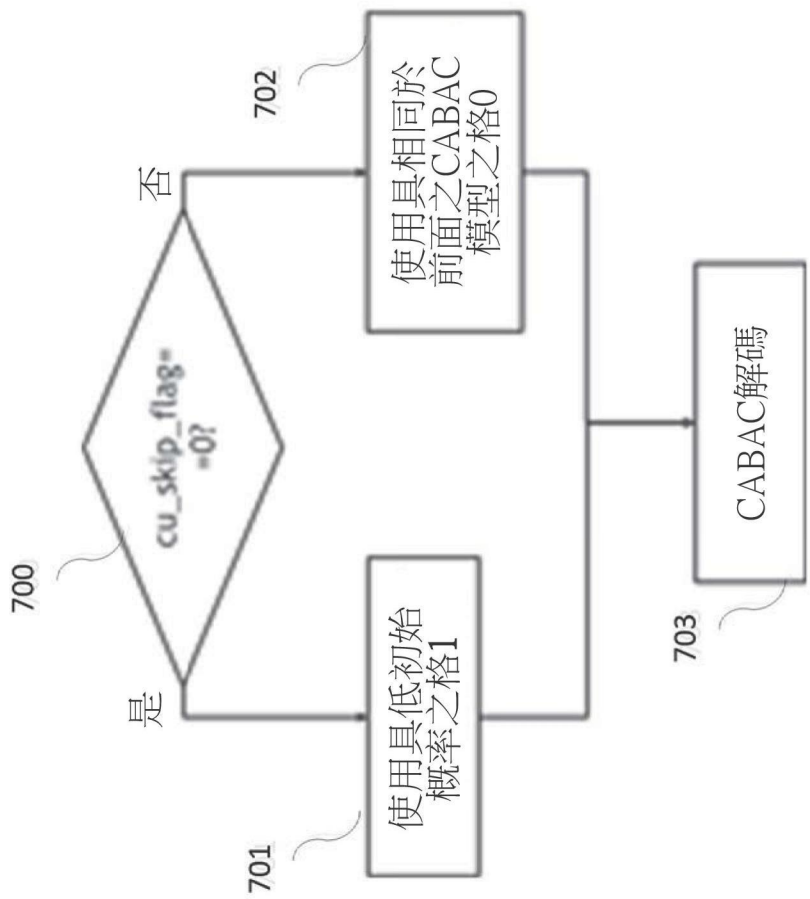
【圖15】



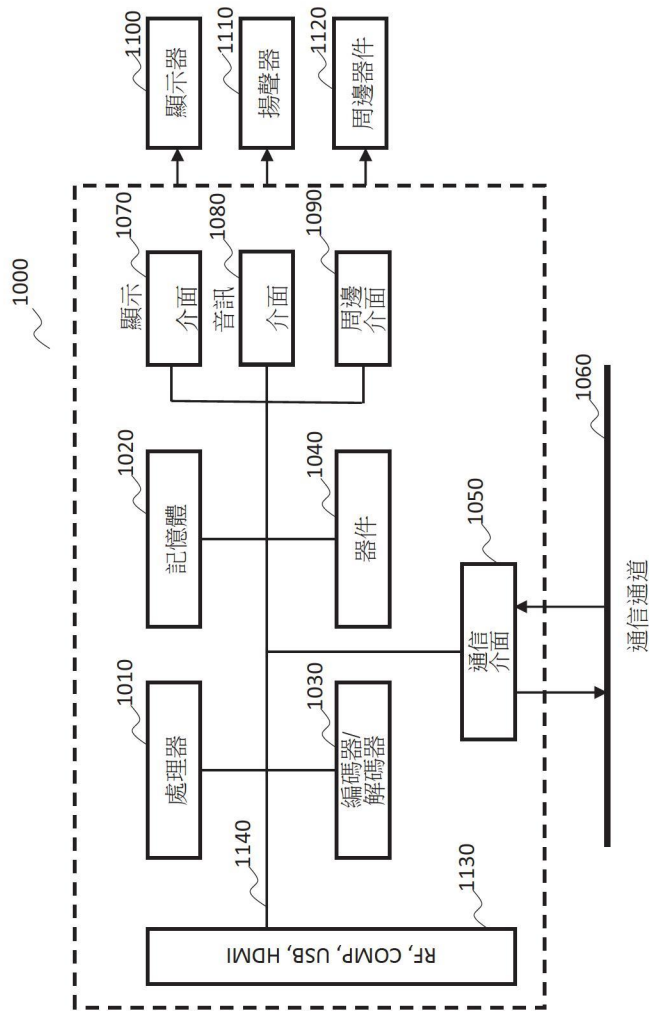
【圖16】



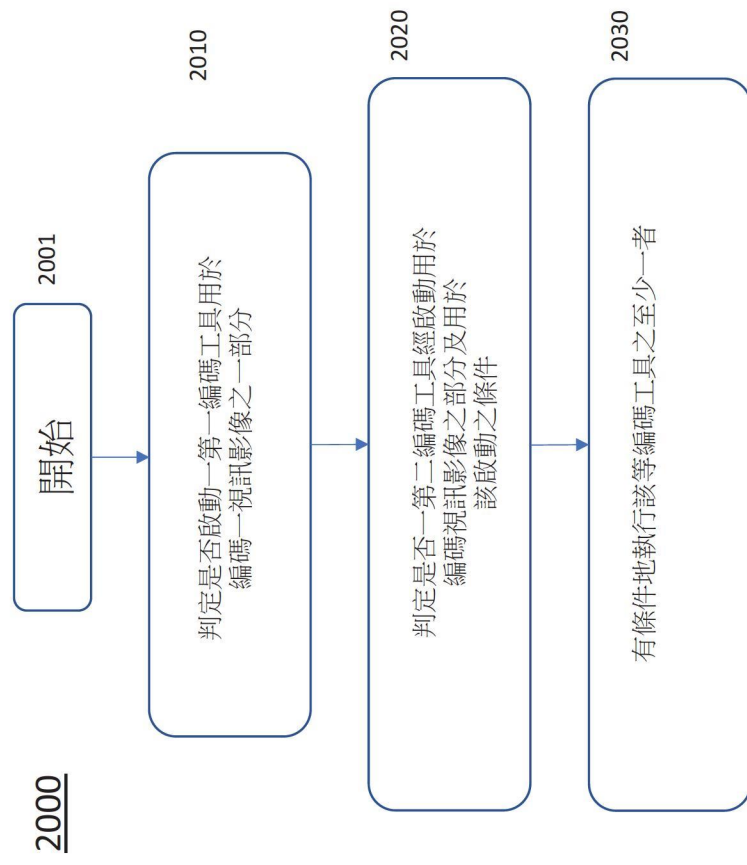
【圖17】



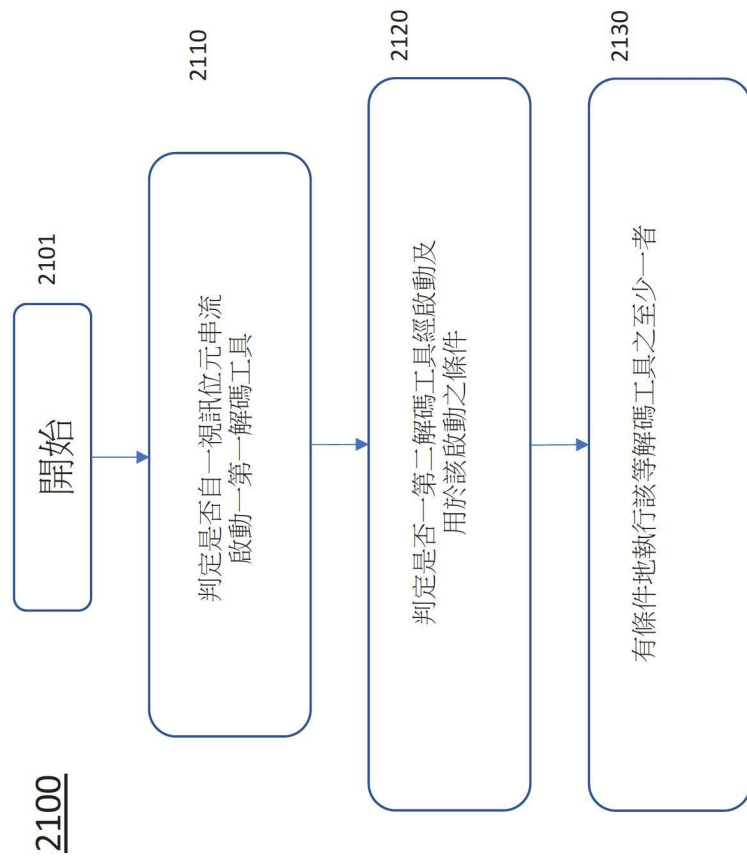
【圖18】



【圖 19】



【圖20】



【圖21】

2200

